



THE UNIVERSITY *of* EDINBURGH

<b>Title</b>	Examining the development of an expert system for reliability data analysis with object oriented principles
<b>Author</b>	Al-Doori, Mulhim
<b>Qualification</b>	PhD
<b>Year</b>	1996

Thesis scanned from best copy available: may contain faint or blurred text, and/or cropped or missing pages.

Digitisation notes:

- Page 136 missing in original

**THE UNIVERSITY OF EDINBURGH  
DEPARTMENT OF BUSINESS STUDIES**

**EXAMINING THE DEVELOPMENT OF AN EXPERT  
SYSTEM FOR  
RELIABILITY DATA ANALYSIS WITH OBJECT  
ORIENTED PRINCIPLES**

**BEING A THESIS SUBMITTED FOR THE DEGREE  
OF**

**DOCTOR OF PHILOSOPHY**

**BY**

**MULHIM AL-DOORI**

**SEPTEMBER 1995**





## **DEDICATION**

I dedicate this work to the memory of my father

To my mother

To All my family for their love and support

## **DECLARATION**

**I hereby declare that this work is my own and I am solely responsible for it.**

## **ABSTRACT**

The work reported in this theses aims to explore the possibility of designing and building an expert system for statistical analysis. In particular it focuses on Reliability Data Analysis.

Expert systems are Artificial Intelligence systems based on Advanced Information Technology which incorporate knowledge-based systems. They require high level problem solving strategies used by experts in consultation.

Several approaches to the building of an expert system are considered. The final choice selected was based on considerations concerning the knowledge domain and reasoning processes and was based on Object-Oriented principles. A Prototype was implemented in APL2.

As a background to the development of an Expert System for Reliability Data Analysis it reviews statistical expert systems including KENS and GLIMPSE. Given the desire to build a system on Object-Oriented principles a brief introduction to this field is given highlighting the major features, including examination of languages such as SMALLTALK, C++ and APL2.

An evaluation of different design approaches is made. This is followed by a description of the system ARDA, APL2 Reliability Data Analysis. This description include steps in the construction, structure, concepts and facilities.

The final chapter of the thesis describes suggestions for further work.

# **CONTENTS**

## **CHAPTER ONE                      INTRODUCTION**

1.1 Introduction .....	1
1.2 Previous Work .....	5
1.3 The Envisaged Research .....	8
1.4 The Survey .....	10
1.5 Outline of Thesis .....	12
1.6 Discussion .....	13

## **CHAPTER TWO                      STATEMENT OF PROBLEM**

2.1 Introduction .....	15
2.2 Expert Systems .....	17
2.3 Statistical Systems .....	20
2.4 Reliability Data Analysis .....	20
2.5 Investigating The Implementation .....	22
2.6 Elements of Reliability Data Analysis .....	23
2.6.1 Data .....	23
2.6.2 Models .....	24
2.6.3 Objectives .....	26
2.7 Statistical Analysis Strategies .....	26
2.8 Discussion .....	29

## **CHAPTER THREE                      STATISTICAL EXPERT SYSTEMS**

3.1 Introduction .....	30
3.2 General Review of Statistical Expert Systems .....	30
3.3 Expert Systems Using Statistical Inference Engines .....	32
3.4 Expert Systems Assisting or Enabling Statistics .....	34

3.5 Other Statistical Expert Systems .....	38
3.5.1 KENS .....	38
3.5.2 GLIMPSE .....	40
3.5.2.1 The Structure of Glimpse .....	41
3.5.2.2 Using Glimpse .....	42
3.6 Discussion .....	42

## **CHAPTER FOUR            PHILOSOPHY OF SYSTEM DESIGN**

4.1 Introduction .....	44
4.2 Hierarchical Systems .....	44
4.2.1 The Structure of The system .....	45
4.2.2 Activities of The System .....	46
4.2.3 General Discussion .....	49
4.3 Frames .....	49
4.4 Frames and Objects .....	51
4.5 PROLOG Based Systems .....	53
4.6 Discussion .....	54

## **CHAPTER FIVE            OBJECT ORIENTED PROGRAMMING**

5.1 Introduction .....	56
5.2 Definitions .....	57
5.2.1 The Object .....	59
5.2.2 Abstraction .....	59
5.2.3 Encapsulation .....	60
5.2.4 Independence and Persistence .....	60
5.2.5 Homogeneity .....	61
5.2.6 Inheritance .....	62
5.2.7 Polymorphism .....	62
5.3 Review of Definition .....	63

5.4 Advantages and Disadvantages of Object-Oriented Programming.....	63
5.5 Practical Issues .....	65
5.6 Comparison of Languages .....	66
5.6.1 CLOS .....	66
5.6.2 SMALLTALK .....	67
5.6.3 C++ .....	68
5.6.4 Discussion .....	69
5.7 APL2 As An Object Oriented Language .....	70
5.8 Discussion .....	72

## **CHAPTER SIX                      SYSTEM'S DESIGN**

6.1 Introduction .....	73
6.2 Desired Features of The System .....	73
6.3 The Design Structure .....	77
6.4 Elements Required in The System .....	79
6.4.1 Data .....	80
6.4.2 Models .....	81
6.4.3 Objectives .....	83
6.4.4 Control .....	84
6.4.5 Knowledge .....	86
6.4.6 Guidance .....	87
6.5 Discussion .....	88

## **CHAPTER SEVEN                  THE SYSTEM**

7.1 Introduction .....	90
7.2 ARDA .....	90
7.2.1 Background .....	90
7.2.2 Description of ARDA .....	91
7.3 Statistical Elements of ARDA .....	92

7.3.1 Statistical Advisor .....	92
7.3.2 A Set Of Statistical Models .....	92
7.3.3 Survival Analysis .....	93
7.3.3.1 Proportional Hazards Model .....	94
7.4 The Structure of ARDA .....	95
7.4.1 Data .....	95
7.4.1.1 Data Definition .....	96
7.4.1.2 Data Validation .....	97
7.4.1.3 Data Input .....	97
7.4.2 Models .....	97
7.4.2.1 Data Model Selection .....	100
7.4.2.2 User's Desired Model Selection .....	100
7.4.3 Objectives .....	103
7.4.3.1 To Repair or Replace .....	103
7.4.3.2 To Calculate Data's Statistics .....	104
7.4.3.3 To Test For Trend .....	105
7.4.3.4 To Fit a Distribution to Data .....	105
7.4.4 Control .....	105
7.4.5 Knowledge .....	108
7.5 A Typical Session with ARDA .....	110
7.6 Summary .....	117

## **CHAPTER EIGHT          FURTHER WORK**

8.1 Introduction .....	118
8.2 Summary .....	118
8.2.1 Concepts With Statistical Expert Systems .....	118
8.2.2 Implementation of Statistical Expert System .....	119
8.3 Further Development .....	120
8.3.1 ARDA's Front Ends .....	120
8.3.2 Natural Language Development .....	121
8.3.3 Interfaces .....	121

8.3.4	Data .....	121
8.3.5	Models .....	121
8.3.6	Objectives .....	122
8.3.7	Help and Guidance .....	122
8.3.8	Explanation Facility .....	123
REFERENCES .....		124
APPENDIX ONE	User Guide .....	135
APPENDIX TWO	The Questionnaire .....	138
APPENDIX THREE	Numerical Examples .....	147
APPENDIX FOUR	ARDA IN CRYSTAL.....	158
APPENDIX FIVE	ARDA IN APL2.....	159
APPENDIX SIX	Published Papers.....	160



### **ACKNOWLEDGEMENT**

I would like to express my gratitude to Dr. J.I. Ansell for his continuous supervision, guidance and extreme tolerance. I am also grateful to Dr. John Glenn for his much valued opinion and support. I would like to thank Mrs. Kit Gardener for her generous help and support to me during my work in the department. I would also like to thank the secretarial staff for their help.

## **CHAPTER ONE**

### **INTRODUCTION**

## **1.1 INTRODUCTION**

Since the beginning of the industrial revolution man's eternal aim to improve the style and standards of living took a sharp turn towards the production of better, faster, cheaper and more reliable products. This desire which manifested itself in the introduction of new technologies in a vast variety of production fields supported by a revival in scientific research resulted in massive expansion and progress in industrial productivity in the western world. This advancement in technology and the process of mechanising life had and is still having a significant affect on life style in western societies making the world more complex. The evolution of a variety of new elements affecting life in general and industry in particular such as the increase in competition, systems are becoming more complex and the need to process vast amounts of information quickly and accurately reduced the manager's ability to control and optimise production. Therefore there was a need to define specific rules that will govern levels of production and quality of products. This lead to the introduction of control standards for availability, reliability and maintainability performances of products and the performance of the maintenance support system. These standards help achieve compliance with contractual obligations. Adherence to international standards which is regarded as imperative for business world-wide.

Often in reliability analysis an attempt is made to predict the performance of large and complex systems which are composed of many components. This is achieved by examining data either from experiments or field conditions to assess the distribution of the lifetime of a system and its related properties. Data are collected and analysed during all phases of a product lifecycle in order to improve, evaluate and verify reliability. Therefore statistical methods play an important role in reliability standards.

One of the problems that faces statisticians in general is the lack of statistical approaches for selecting appropriate characteristics and designing more efficient test surveys or experiments. Revised and new procedures provide algorithms to

replace tables and monograms. These algorithms are intended to facilitate the user in developing computer based evaluation tools. Thus the rough indicators arrived at by traditional means are exchanged by more flexible technology. One statistical area that can benefit from the introduction of such approaches is reliability data analysis. The compliance with contracted reliability requirements, while necessary, is not sufficient for successful business. The improvement of reliability testing facilities is a positive move towards better business. The introduction of procedures is aimed at allowing the user to exploit the power of information technology in planning tests and analysing the resulting data. Recent changes in the structure of certain International Electrotechnical Committee (IEC) standards provide an opportunity to present appropriate, more efficient data analysis and inferential procedures to the reliability engineer. This allows reliability tests to have measurable performance characteristics and be carried out under repeatable and reproducible conditions. There is a need to develop a system that will encapsulate reliability knowledge and facilitate with ease statistical analysis. The idea of an expert system lends itself to such complex and specialised domain as reliability analysis.

Expert systems are a development of advanced information technology being a part of artificial intelligence research. They are knowledge based systems which aim to employ knowledge to simulate the behaviour of human experts. An expert system uses knowledge allied with inference to solve problems which are difficult enough to require significant human expertise. There have been a large number of systems built which claim to be expert systems. They have been implemented in a number of forms both in terms of languages used and the inferential bases on which they have worked.

The academic success of expert systems since mid 1970s has perhaps hidden the fact that a major market for such systems did not begin to emerge until recently. Indeed many of the most widely publicised and best known expert systems have only been used in an experimental mode, or had their use discontinued. Projections of the size of expert systems future market vary but most suggest that it will be

huge. Expert systems report in 1987 suggested a growth in US market for the development of expert systems from \$274 million in 1986 to \$1,767 million in 1992. Goal (1994) suggests that by 1999 US companies will be spending 15% to 20% of their IT budget on expert systems. This however is dwarfed by the size of the market for knowledge itself, and as the knowledge component rises the rate of growth in systems will increase.

Having been relatively successful in technologies and scientific disciplines such as medicine, chemistry, engineering and military endeavours, expert systems moved gradually into management decision making. This deals with managerial decision making for multi-disciplinary problems, and hence additional measures needed to be taken when designing management expert systems.

An expert system will promote an understanding of the uncertainties associated with the process of selecting, applying and interpreting statistical information collected during reliability testing. It will improve the users' knowledge of reliability analysis, produce repeatable tests and improve communication between engineers and statisticians through reliability tests. Thus it is hoped that the adaptability of an expert system should introduce some creativity that is good for both the Statistics and reliability engineering.

Expert systems are a development of advanced information technology being a part of artificial intelligence research. They are knowledge based systems which aim to employ knowledge to simulate the behaviour of human experts. An expert system uses knowledge allied with inference to solve problems which are difficult enough to require significant human expertise. There have been a large number of systems built which claim to be expert systems. They have been implemented in a number of forms both in terms of the languages used and the inferential bases on which they have worked. The objective of this thesis is to explore the development of expert systems for statistical analysis. The specific area of study is reliability data analysis. In the work several key features are established which should aid further researchers.

As part of the study it was thought appropriate to examine the aspects of software engineering which impinge on the development of the system. There is a desire to produce software which is capable of fulfilling the requirements of the system but also software which is reliable in an efficient manner. There has been for sometime a research overlap between artificial intelligence and software engineering with the use of acquired knowledge to produce improved systems. In the current work the concepts of object oriented programming have been found to provide an efficient way of developing reliable software.

The central theme in artificial intelligence is the attempt to encapsulate the knowledge of an expert in a computer system effectively. Successful development of an expert system requires the encapsulation of the knowledge of an area of application along with the skills to manipulate the knowledge which usually follow the meta level problem solving strategies used by the expert. Such a process is multi-disciplinary. The development of an expert system in a specific domain requires the comprehension of the knowledge of the domain but also the reasoning processes of the domain. Hence in the development of the expert system there is a need to systemise the subject area and formalise the reasoning process. It was therefore necessary in the current work to provide a framework for the knowledge domain and explore the processes of decision making.

Obviously within the limitation of a doctoral thesis it is not possible to develop a fully a commercial system. The objective must be more limited and in the case of the work reported in this thesis the objective has been to demonstrate the solution of many of the problems involved with the development of an expert system for statistical analysis especially in the are of reliability data analysis.

Expert systems popularity increased in the eighties as a decision support tool for managers. The main aim behind developing an expert system is to reduce the cost of decision making by increasing the efficiency but also raising the level of accuracy. An expert system should provide a sound , consistent and more rational

decision making mechanism which may replace the expensive expert. Expert systems should provide better documentation of the rationale for the decisions.

To provide a background for the research a brief history of expert systems have been produced. The main trends and developments within expert systems have been explored. The outcome of this history is to highlight both the diversity of thought about expert systems and the slowness in development of successful systems.

As a start point for the development of the current work a survey of people involved with reliability analysis was carried out. The objective was to gain an insight into their views on expert systems in general terms. The aim was not to treat those surveyed as clients for whom the system would be developed since it was felt that this would bias the design of the system to a small self-selecting group. The survey provided awareness of the users' comprehension of expert systems and their views of them. The survey provided a useful basis on which to develop the system and helped in forming judgement on the design of the system.

## **1.2 Previous Work**

Although expert systems have received a lot of attention in recent years the ideas behind them are not new. Mankind's interest in employing machines to think and make decisions for them goes back a long time. Concentrating on the last fifty years a significant contribution was made by Turing in the areas of grammars and languages, Ansell (1987). The impetus for the birth of artificial intelligence in the western hemisphere was the second world war. In 1943 Warren McCulloch and Norbert Weiner used cybernetic ideas and founded neural-net theory, see Forsyth (1984). They worked on a limited problem concerning a frog's visual system. These ideas were the base for the work of Rosenblatt (1957) who created PERCEPTION. This system was capable of recognising a limited class of patterns, see Minsky and Papert (1969).



In the 1960's Newell and Simon following on from work started by Turing and Shannon developed GPS, The General Problem Solver, see Forsyth (1984). The aim was to design a system which could be used to solve problems of different types from a variety of application areas. This system developed several concepts which have become a standard part of expert systems terminology. "Task Environments" or application domain were to be specified by the user. The system then reduced the subject matter to a series of smaller problems. The method employed was "Depth First Search" which is a search strategy in which one path of possibilities is developed and checked at a time backtracking to develop other paths. The work of Newell and Simon was based on heuristic search. They regarded solving problems as a search controlled by heuristic rules to decide on the right solution. The method employed can be regarded as an inefficient procedure since it does not look at all possibilities thoroughly. Problem solving in AI adopted the idea of "State-Space-Search" which involves a search through a network of nodes, a node for each problem state. Using this approach with any real life field was difficult because the state space would tend to be very large. It became difficult to obtain solutions through an exhaustive search of space.

One method of dealing with large search spaces is called "Generate-And-Test". This method was used in DENDRAL a system developed at Stanford University to elucidate the structure of compounds in Organic Chemistry, see Alty and Coombs (1984). The project started in 1965 with the aim of producing computer aid to chemists who lacked expertise in certain analytical techniques. It was so successful that it encouraged the development of other Expert Systems.

MYCIN by Shortliffe 1976 was developed at Stanford University to support physicians with the diagnosis and treatment of blood bacterial infections. It laid the blue print for Expert Systems. Shortliffe based his work on the theories of Minsky and McCarthy (1974) on knowledge representation to produce the system. It was improved with the introduction of EMYCINE, see Shortliffe and Buchanon (1975). It was also noteworthy that the field of application of MYCIN was Medicine. This has proved to be a field which attracted many workers in Expert



Systems. The reason is the ability to classify symptoms and diseases into groups with ease, see Duda, Gashing & Hart (1979).

Another development in the 1970's was PROSPECTOR developed by SRI International in association with consultants and the U.S. Geological survey. This system was designed to assist geologists to evaluate sites for the existence of certain deposits. In PROSPECTOR highly specific models are coded into the computer system. These models are used by geologists when testing for ore deposits. The system is interrogated to give site evaluation advice. The knowledge is stored in sets of facts and rules. A model frame is connected by rules where a frame could be some evidence, hypothesis, or fact with special attributes of facts.

An early attempt at mimicking experts was INTERNIST system (1974) developed by team working at Pittsburgh. It simplified the structure of a diagnosis to lists of possible disease and symptoms. From the set of symptoms it is possible to evoke a set of diseases and from diseases it is possible to manifest some manifestations or symptoms. INTERNIST used mainly associational relations between manifestations and diseases and does not attempt to model disease processes. It was improved in the development of the system CONSTRICTOR which reduces the range of target disease areas.

In the 1980's systems have become more powerful especially in specialised areas, for instance MACYMA was recognised as a powerful mathematical system for aiding with mathematical analysis. This has been improved in MATHEMATICA. Lenat made a machine learning system called EURISKO, Lenat (1982). This system automatically extends and improve its heuristic rules. Specific systems were made to do specific jobs with the development of database new methods were used like frames and blackboards.

This brief history of expert systems clearly establishes the length of time it has taken to produce the concept behind expert systems. It also demonstrates that developments have arisen not necessarily from a consistent coherent approach to

research but has been more opportunistic. Development have often occurred from systems designed for new areas where there has not been a well established frame work.

### **1.3 The Envisaged Research**

The objective of the study reported in this thesis was to explore concepts in the production of expert systems. To provide a focus for this study the possibility of building an expert system for statistical analysis. Obviously this would be too broad a field to explore within the limits of a doctoral research programme. Therefore attention has been restricted to building of an expert system for reliability data analysis.

The choice of domain was influenced by the lack of an expert system for reliability data analysis, the wealth of literature about reliability data analysis and the reasoning processes lent itself to being implemented. Whilst there is a wealth of literature on reliability data analysis it was felt the area was sufficiently clearly defined to be contained within an expert systems. This will be explored further in a subsequent chapter.

The building of an expert system for a domain requires the systemising of the knowledge of the domain and also a formalisation of the reasoning processes used within the field. It was therefore necessary to characterise the reliability data analysis. Firstly there was a need to define the area of reliability data analysis. Therefore a boundary for the subject had to be established. Having established this the next stage would be to consider the reasoning approach employed. From the literature the impression arises of a technique lead approach to analysis, mainly since the aim of authors is to introduce new techniques. In analysis, however, the approach will be problem lead and so there was a need to provide a mechanism which would take such an approach.

The production of any software requires a number of decisions to be made such as the language to be used and philosophy of design to be employed. Given the aim to explore the production of expert systems this formed a central part to the work. The first question is to resolve whether one can use a expert system shell or a language. A shell provides tools to assist the developer. The main disadvantage, though, is the restrictions imposed by the shell. Generally the lack of flexibility will not allow the user to appropriately develop the system to meet the requirements of the user. The alternative using a language can be equally problematic. AI languages such as PROLOG, LISP and C++ will also restrict the user and there will have to be development of utilities to meet the requirement of the user. Similar restrictions arise with more traditional languages PASCAL or FORTRAN.

From the survey, which is described in the next section, it was obvious that most of the respondent were using expert system shells rather than developing from languages. Most of the systems they used were either rule based or frame based. In the initial stages of the research effort was expended on the use of an expert system shell, CRYSTAL. This was chosen since at the time of the start of the research it was a system which had achieved some acceptance. It was a rule based system with a tree like structure. This route for development of the expert systems proved abortive since it did not appear capable of achieving the type of system required. The work is reported in chapter three.

Subsequent to the work on CRYSTAL it was felt appropriate to explore the use of a language for the production of the system. Given the outcome of the work on CRYSTAL it was appreciated that a language capable of dealing efficiently with statistical algorithms was required. Whilst there are a large range of contenders APL2 was chosen since it provided a range of useful facilities.

Beyond the choice of language there is the question of philosophy of design. There is a need to produce software which is reliable in an efficient manner. There are a wide range of possible programming approaches that can be taken, from structured programming which interprets the requirements in a formalistic manner

into a software to more liberal approaches such as Objected Oriented Principles. Given the desire to retain the ability to introduce innovatory concepts into the software it was felt the structured approaches could not be taken. The modularity of Object Oriented Design was attractive since it seemed most appropriate in the context of the current work. It was necessary therefore to explore Object Oriented Principles which are covered in chapter five.

The objectives of the research can therefore be restated to be:

- 1) Exploring concepts within statistical expert systems.
- 2) Investigating the production of statistical expert systems.

To achieve these objectives it was decided to build an expert system for reliability data analysis. This would include the systemisation of reliability data analysis and the process of analysis. It would also require the investigation of software in which to implement the system as well as exploring the principles of design.

The outcomes expected of the research are therefore developments within the area of statistical expert systems, a systematic view of reliability data analysis and the implementation of the developments in software. It is not expected that the software would reach a commercial level production, but would be a prototype system which demonstrated the possibility of building a full system. There will be opportunities for further developments of the system beyond the scope of this thesis.

#### **1.4 The Survey**

To gain insight into views of individuals in Reliability about statistical packages and expert systems a fact finding survey of market place was attempted, see Appendix 2. The objective of this survey was to set the scene for the research gaining an understanding of the potential end users. Views were sought on their experiences of systems and of the development of systems.

It was not the aim of the survey to gain from the potential end users a specification for the system. It was felt that gaining a specification would entail considerable commitment from the end users which could not necessarily be guaranteed. There would have had to be repeated questionnaires or visits to the group involved to clarify the specification. This would have taken sometime but would not necessarily have produced a specification that could be built. There was also the possibility that the individuals who responded and were willing to offer the commitment may not be typical end users. This may bias the system. For these reasons it was felt that the approach could not be followed in this research.

The survey covered about 250 potential end users of statistical systems in industry. They were selected from a variety of industrial organisations and research centres who were actively involved in the development and use of statistical software in particular the area of reliability of data. They were surveyed using a postal questionnaire using pre-paid envelopes.

The questionnaire was in two parts: a general section to explore common software problems in industry and a more specific survey on particular aspects of expert systems.

The response rate was 26%. The total number of responses was 66. This was achieved without following up those surveyed. Whilst this is a good response rate for a postal survey the total number of responses cannot be regarded as sufficiently large to come to strong conclusions. Given the time of the survey there was limited response to the questions concerning expert systems. Again this restricts the information that can be derived from the survey. Hence the responses can only be treated in a qualitative rather than a quantitative manner, ( see appendix 2 for details of the survey's questions, users and companies ).

The survey's answers can not be regarded as reliable due to the low number of users replying but they are worthy of taking some interest and give a general feeling of the problems faced by expert systems users. From the survey it would

appear that users faced two main problems, the first is the lack of economic benefit of the expert systems currently used. The performance of expert systems did not justify the allocation required of system development effort. The second is the little success current expert systems enjoyed in providing results accurate enough to rely on in decision making.

### **1.5 Outline of Thesis**

The aims as stated in section 1.3 are:

1. Exploring concepts within statistical expert systems.
2. Investigating the production of statistical expert systems.

These will be explored through the development of a system to aid reliability data analysis. Chapter two will be a statement of the problems to be faced in tackling these objectives. There will be a general discussion of what constitutes an expert system. A definition of reliability data analysis will be presented along with some of the basic problems of the area. There will be an examination of the strategies used in statistical analysis. There will be a brief review of the approach taken to fulfil the objectives of the research.

Chapter three will provide a review of statistical expert systems highlighting the perceived developments within the area. It will particularly examine two expert systems which seemed to be relevant to the current project. The initial attempts at building an expert system using an expert system building shell will be described in chapter four. Other system design approaches are discussed in chapter four such as frame based systems and PROLOG based systems.

Given that it was decided not to build a system using a shell it was necessary to review the process of production. The review lead to the decision to use object oriented programming as an efficient way of producing reliability software. In chapter five object oriented programming is discussed and several object oriented



languages are reviewed. The desire, however, was for a language which would aid the development of statistical routines as well as using object oriented principles. Whilst APL2 satisfies the former it is not traditionally viewed as an object oriented language. Therefore the later part of the chapter is devoted to an exploration of whether APL2 can conform to object oriented principles.

The detailed requirement for the system are presented in chapter six. This explores the implementation of the design decision and describes in more detail the mechanisms of the system. Chapter seven considers the details of implementation and highlights the features of the system. To give the reader a view of the system a typical session is presented in chapter seven.

In the last chapter, chapter eight, there is a summary of the achievements of the research work. Also explored within this chapter is the areas for further work.

## **1.6 Discussion**

Expert systems are a collection of knowledge and procedures to categorise that knowledge based on human knowledge and experience, often termed 'know-how'. The advantages claimed over humans are they are permanent, more consistent, speedy, accessible and expandable. The progress in Artificial Intelligence research has been directly related to the funds available. Having survived the seventies first winter, see The Lighthill report (1972), Artificial Intelligence enjoyed a lease of life in the ALVEY initiative. When the JIPDEC (Japanese Information Processing Centre) heralded an advance in the computing fifth generation, researchers in the U.K. both in industry and academics worked jointly creating an immediate growth and greater involvement in Artificial Intelligence research. This produced what might be called the first generation of AI software which was the result of experimenting with new principles of programming. The reaction of industry was disappointing and a lot of confusion and criticism surrounded these products.

Early work in expert systems gave a positive sign to encourage industrial and academic research centres to invest in the development of more sophisticated and complex expert systems. Several organisations began to take on board expert systems in their different processes. This enthusiasm dropped gradually due to the amount of effort required in building expert systems and the cost of developing them. The lack of economic benefit of some of the expert systems produced is forcing expert systems into another winter. The industry mood towards expert systems was summed up rather well by the comments of a respondent to the survey “ We co-operated in general development of system in Q5 as part of the ALVEY initiative. The task appeared relatively simple and ideally suited to an expert system. Eventually we had to conclude that it was not cost effective in terms of resources - the pay back period will be infinite. It seems to me that the relationship between sophistication and resource is exponential or worse. Our need for precision put it out of our reach”.



## **CHAPTER TWO**

### **STATEMENT OF PROBLEM**

## **2.1 Introduction**

Expert systems have been an active field of research over the last three decades. This has resulted in a wide range of systems which claim to be expert systems employing a variety of strategies. Many applications of expert systems have been considered but few have been successful. The failures though also have been fruitful in highlighting the possible problems when attempting to codify areas of expert knowledge and then incorporate it into a computer system. The problems are usually related to size and tractability. Many areas are too large to be successfully encapsulated within a system. In other areas there are problems associated with the type of knowledge and how it may be manipulated. An expert system can only be feasible if the area it is defined for is sufficiently limited and well defined.

An expert system is a software system which applies artificial intelligence to behave like a human expert, checking complex situations against a data bank of possibilities. While other types of software systems on computers do calculations, it reasons. This requires a complex language, the capacity to recognise particular situations and variations of them in real world, and the ability to reason from information to possible actions, comparing possible actions and their likely outcomes with a desired outcome, in order the best possible action can result. This will also require the software to be able to choose what will count as the best possible result when faced with a range of likely outcomes. In other words, an expert system will make two kinds of judgement, firstly in recognising a situation, and secondly in choosing the best (or least worst) possible action.

Expert systems in management involves the production of facilities to assist management's decision making for multi-disciplinary and complex problems involving many behavioural variables, and hence additional measures needed to be taken when designing management expert systems (MES). The expert system will provide knowledge and reasoning procedures while the decision-maker will supplement it with the overall problem solving strategy. Reliability data analysis is a well defined domain ,” the study of reliability engineering is concerned with the improvement, analysis, assessment, and prediction of the performance of equipment and systems ”, see Ansell and Phillips (1994). The aim of expert systems is to be able to tackle ill-structured and sufficiently complex problems.

The practical problems encountered in reliability data analysis requires the definition of strategies that will influence the analysis process. This lack of analysis strategy is due to the complexity of the subject and can be resolved within the reasoning activities of an expert system.

Decision support systems, DSS, emerged to tackle problems of less complexity than reliability of data. Although they have planning, analysis, searching and trial and error capabilities. But these capabilities are of a limited nature and will require the intervention of the user to influence reasoning processes and choices far more than what an expert system requires and the final decision is made by the user. In reliability analysis engineers and managers will not want to be involved in every step of the system's decision making process. The complexity of calculations in reliability data analysis will only be clearly understood by a statistician with experience in the subject. Therefore the user will need to be at a different level of processing than the analysis machinery of the system. Therefore it is more logical to design the system as an expert system rather than a DSS.

The main objective of the research is to explore the development of expert systems for statistical analysis. There have been a number of expert systems developed for such. Two system which are particularly notable are KENS, Hand (1987) and GLIMPSE, Nedlder (1989). KENS follows a hyper-text approach to aiding the individual to understand an area in statistics. GLIMPSE assists the user to employ a statistical technique. Both will be described in more detail later in the chapter since they provide different views of the approach that can be taken. Beyond these few statistical systems have gained general acceptance. There are a number that provide to specific users with particular expertise in restricted areas.

Since it is not possible to contemplate building a general system for statistics it was decided to consider an area of statistics which was sufficiently self-contained. For this reason reliability data analysis was chosen. The aim will therefore be to develop a system which offers the user statistical advice and assistance in the reliability data analysis. To the author's knowledge, at the time of inception of the project, no expert system existed for this area. The lack of systems might imply that reliability data analysis is not amenable to the development of expert systems. This could either arise from the area being ill defined or the reasoning process

being such as not to be easily implemented on a machine. Hence there is a need before building a system to demonstrate that the area is well defined and the reasoning process can be developed within the current computer technology. This means that it must be possible to specify the area clearly and describe the main features of the area. The knowledge domain of the area needs to be defined which implies it must be possible to systemise the area of knowledge. It is also necessary to be able to describe the reasoning process and demonstrate it is capable of implementation.

Having decided the scope of the expert system there is a need to explore the production of the system. This implies an investigation of the possible means of production. This implies a series of choices have to be made. There are questions about the type of expert system to be used. There is the need to decide whether to use a language programming or alternatively use an expert system shell. Dependent on these decisions it may be necessary to explore the philosophy of design. Given the second objective of the research is to investigate implementation a variety of approaches will be considered and some will be implemented.

In the development of the system an emphasis is laid on giving the user the facility to choose and control the steps of the analysis. This, when achieved, will give the user the satisfaction of realising exactly what he/she wishes to achieve without having to follow a set routine of operations and processes. In most current expert systems the user does not regularly have this facility and usually ends with unrequired information and longer than necessary, or desired, sessions.

Given the successful development of concepts for building the system the final objective of the project is to consider whether the elements of such a system can be exported or generalised to other areas of statistical or reliability analysis.

## **2.2 Expert Systems**

Expert Systems are problem solving programs that solve substantial problems generally conceded as being difficult and requiring expertise. By building expert

systems in domains like Statistics, Medicine and Law researchers hope to gain insight into the way the people apply expertise. They originated from traditional data processing and are the result of continuous research into the automation of human information processing. Knowledge processing within expert systems provides the facility to control inference which were specific to a particular knowledge domain.

Expert systems operate on two levels of information processing. A high level where development work is done, and a low level where the user works. The higher level is the abstract problem solving stage, processing information to produce knowledge. Construction of rules, construction of guidance, thinking about what form to adopt to represent knowledge in the system and developing the structural design of the system. The lower level is the operational level processing the user's inquiry and manipulation of data.

There are several approaches to represent knowledge in expert systems which include semantic networks, objects, rules, frames and logical expressions. Each method has advantages and disadvantages. A knowledge base is formed having chosen the appropriate approach to encode knowledge and interpreted that knowledge into the selected format. This knowledge base is independent from the reasoning strategy and data. Therefore an expert system's pool of knowledge can be updated and expanded by introducing new data and combining both the user's operational knowledge and further familiarity with the system. Another important design factor at the development level is the analysis strategy which influences the path of analysis and therefore the system's flexibility.

There are several reasons for the choice of domain. Obviously the problem has to be sufficiently complex to be of interest, it should not be a simplistic model. Therefore it has got to be a developed knowledge domain with sufficient literature. The types of problems likely to be considered by the user should be structured or semi-structured. Also there needs to be access to expertise. For these reasons analysis of reliability data seemed an appropriate area to study.

Jackson (1991) suggests that an expert system can be distinguished from a more conventional application program in that 'in building an expert system the chosen domain has to be of evident complexity and normally needs a substantial amount of human knowledge'. Jackson then restricts expert systems design to solve problems by 'approximate or heuristic methods'. These are different from algorithmic solutions in the uncertainty of their success. This restriction to design technique has not been echoed by other expert systems' developers, indeed several expert systems have been designed to depend in the reasoning processes on algorithmic techniques.

Jackson emphasises the point that an expert system should mimic the expert in following the same steps in solving a problem and achieve similar if not more coherent results. This opinion can also restrict the flexibility of an expert system since it does not allow the system designer to take advantage of certain techniques and technologies available in expert systems but not used by the expert. As mentioned earlier the main aim of an expert system is to solve complex problems. Therefore by the introduction of techniques or by making choices not normally used by the expert the complexity is reduced then these methods should be used rather than those of the expert.

Further to processes such as data retrieval and numerical calculations, Jackson suggests that an expert system should represent human expertise in its reasoning. Codifying expertise is normally separated from reasoning code in an expert system. Jackson also stresses the importance of explanation facilities and transparency in expert systems since they may be used by a wide range of users rather than only their creators.

Whilst some of Jackson views are sound on the need for explanation and the need to incorporate the 'reasoning processes' of the system there has to be some debate on the other aspects he suggests. Some human decisions are arbitrary and inconsistent. By applying an expert system one would hope that decisions were not arbitrary and consistent. A system may also need to incorporate the user in

the decision process. A user may well wish to be advised on an issue but not necessarily forced to take that advice. It is therefore the desire within this research project to develop a system which allows the user to take control of, responsibility for, the analysis. This is seen as a key feature.

### **2.3 Statistical Expert Systems**

There have been numerous statistical expert systems. Statistical expert systems are divide into two main groups. The first are those that use statistics as the inference engine for the expert system as a way of dealing with uncertainty. The second group are those which give advice on statistics. It is the second group which is most relevant for the current systems but the first group may well offer ideas and concepts worthy of incorporation.

As a first stage in the research it is therefore necessary to review the range of expert systems and examine which system(s) offer the best 'blue-print' for the desired system. This does not mean that it is intended to follow the analysis proscribed by others but the aim is to build on previous researchers.

One obvious conclusion about the range of statistical expert systems available at the beginning of the research was that they were focused on techniques rather than on the analysis of a problem. This highlighted the narrow view often taken in statistics that problem solving is purely fitting the appropriate model and interpreting the outcome of fitting the model. It was therefore central to the development of the system that the objective of the analysis should play a central role. Chapter two reviews statistical expert systems and reflects on the types of system which might be employed as a 'blue-print' for the envisaged system.

### **2.4 Reliability Data Analysis**

In building an expert system there is a need to have a clear concept of the knowledge domain. This requires that the knowledge domain should be well



defined. A measure of whether an area is well defined is the literature associated with it. There has developed a considerable literature on reliability and a significant portion of this is devoted to reliability data analysis. A rich literature whilst being useful in developing the 'knowledge domain' may be too large or too complex to encapsulate within a system. Reliability data analysis has the disadvantage that it overlaps with many fields in statistics. It shares survival analysis with medical statistics, and uses techniques from other areas such as multivariate and time series analysis. Therefore there is a need to define reliability data analysis sufficiently clearly to allow the development of an expert system.

Reliability is the study of the performance of a system. Particularly it is concerned with whether a system is performing at an acceptable level when required. The objective of reliability is to produce systems so that they achieve the goal of function at the acceptable level when required. Reliability data analysis is the examination of data from either experiments or field conditions which allow assessment of the reliability. It is usually interested in assessing the properties of a system. The main property of interest is the distribution of the lifetime of the system. In this thesis reliability data analysis will, therefore, be interpreted to be the assessment of the distribution of the lifetime of a system and its related properties.

At the time of inception of the project there was not available a software package that would provide all the facilities that were thought necessary. It was therefore decided that the routines required for analysis would be required to be constructed within the system devised. It was therefore realised that some time would have to be devoted to the construction of the appropriate routines. Some of the basic issues of analysis are discussed in section 2.6 of this chapter.

Of equal importance as to whether the area is well defined is the need to consider the underpinning reasoning process. Whilst both the practice of analysis within statistics and its philosophical underpinning have attracted considerable attention, the development of approaches to or strategies for analysis have until recently



unfortunately received less attention. There has been some discussion on reliability data analysis, see Ansell and Philips (1989) and O'Connor (1991). The discussion though has mainly centred on role rather than strategy. In this research there is an attempt to describe a plausible strategy from which it will be possible to build an expert system. This is further discussed in section (2.7).

## **2.5 Investigating The Implementation**

As stated in section 1.3 the second objective of the research is to consider the implementation of the expert system. There are a variety of possible approaches to building expert systems. Some types of expert systems will imply specific types of implementation. For example if a rule based system was considered then one may as well use a MYCIN-LIKE system since this will provide the architecture required. Unfortunately the penalty of using such a system will be high if one wishes to innovate for such systems tend to be fairly constrictive of what can be achieved. They though will generally supply a number of utilities to ease building of the expert system. Therefore there is a trade off between these two aspects. Such systems are often known as expert system shells.

If one is not to use an expert system shell then the alternatives abound. There are a range which have become known as expert system languages such as PROLOG. Such languages again provide a rich medium to develop attributes of an expert system but they lack again utilities which might be deemed helpful for statistical analysis. This can be remedied either by building utilities within the language or alternatively using the expert system language as a front end to another language or package for example GLIMPSE.

Another approach is to employ a conventional language to develop the expert system in. This has the disadvantage that the elements of expert system have to be created in the language. This again maybe far from ideal.

Under any of these options a strategy has to be developed for construction of the software. Obviously this will be effected by choice of type of system and also of the language chosen. It may be necessary to establish that a language can conform to the philosophy of design chosen.

In the research these aspects will be explored. Using an expert system shell will be considered as well as employing an expert system language. Eventually a system will be developed using a conventional language.

## **2.6 Elements of Reliability Data Analysis**

A definition of reliability data analysis was given in section (2.4) of this chapter, however, it is appreciated that such a definition does not fully indicate the nature of the knowledge domain. It would of course not be possible to describe the whole of reliability data analysis within a short section yet it is felt appropriate to pick out some salient points.

### **2.6.1 Data**

A major concern for any statistical analysis is the form of data. In reliability the data which arrives on the analyst desk, tend to be problematic. It is very rarely clean and often needs considerable work before a coherent set of data can be uncovered. Added problems may arise if the data comes from one of the extensive data banks. In the past these have been poorly maintained and organised which frequently lead to false impression of plant or equipment behaviour.

Data rarely arises from designed experiments, though this may be changing. It is often collected as an adjunct to other activities. Hence there are either missing, misreported or erroneous points within the data set. Times tend to be collected on a variety of bases, they may be calendar times, working times (or related measures) or a mixture. Full contextual detail is rarely retained. Some of this may be rectified through the increasing emphasis on data collection. For single shot systems usual

time to failure is recorded or time to system removed from trial. Hence the data is a set of lifetimes possibly with censored data. Types of censoring are discussed in number of texts, see Lawless (1982), and Nelson (1991). Lifetimes are usually denoted by  $x_i$ .

Alternatively data may arise from repairable systems. It is assumed that the systems is maintained so that if a component fails then it will be replaced by another component. For repairable systems a simple model is given is presented in figure (2.1).



Figure 2.1 - A simple model for repairable systems.

Usually there is a sequence of failure/replacement times. In the figure  $n$  observed events have occurred in time  $T$  from a start time.  $t_1, t_2, \dots, t_n$  are the time of these events. The data consists of observations  $t_{i-1} < t_i < t_{i+1}$ . The time between an event is  $t_i - t_{i-1}$  usually denoted  $x_i$  and is the lifetime of the  $i$ th components. Obviously there will be truncation and censoring again if only at beginning and end of period of observation.

As well as the times of events or times to failure, information should be included on context of data. This should include how it was collected where and when. It may also include information on other variables such as design factors or prevalent conditions. These variables will be covariates.

### **2.6.2 Models**

There are a wide range of statistical models which are employed within reliability. For single shot systems the lifetime may be modelled by an appropriate

distributions. McDonald and Richards (1987) describe a two related sets of distributions which cover many of the standard distributions used in reliability. Non-parametric models also can be used to estimate the underlying distribution for example using the Kaplan Meier or Nelson estimator, see Ansell and Phillips (1994).

For a repairable system there is a need to characterise the underlying stochastic process. Ascher and Feingold (1984) list five generic types of possible stochastic process models for repairable systems. These are Superimposition Renewal Process (SRP), Renewal Process (RP), Branching Poisson Process (BPP), Non-homogeneous Poisson Process (NHPP) and Homogeneous Poisson Process (HPP). Kimber (1989) suggested that Branching Poisson Processes were not generally used in reliability. There is the possibility of other models. Other aspects may effect the analysis such as there being a trend in the data or the data being related. It therefore will be important to test for trend and dependency within the data.

If there are available covariates for either single shot systems and repairable systems it will be possible to explore the relationship between the lifetimes and the covariates. This can be carried out using a wide range of models, see Ansell and Phillips (1994) for example using Cox's Proportional Hazard Model, see Cox (1972,1974). This model have been applied with a significant degree of success. The model is semi-parametric in which weak assumptions are made about the hazard. It is a robust model which can show factors affecting data, Solomon (1984).

The application of Proportional Hazard Model to field data has been considered by many authors in reliability such as Bendell and Wightman (1985), Ansell and Ansell (1987), and Jardine and Anderson (1984). Data can often be a repairable process because data often originates from maintenance record which consist of times of events such as repair or replacement on components. Cox's model has been applied to repairable systems, Ansell and Phillips (1994).

### **2.6.3 Objectives**

All analyses should have an objective which may be implicit or explicit. Hence in reliability data analysis we need to define the objectives of the analysis. Deciding on these objectives depends on the reasons for the study. Ansell and Phillips (1994) provides a list of possible general objectives for the analysis of reliability data of a system which can be one or any combination of the following:

- 1- Checking whether or not a system has reached a certain level of performance.
- 2- Estimating the cost of achieving a given level of performance.
- 3- Calculating the likelihood of a task to succeed.
- 4- Deciding on use of resources to achieve the best results.

It is important when executing a statistical analysis to continually review the objectives of analysis. The analysis should be problem-led rather than technique-led. If simple technique can achieve the required objective and produce the required results then there is little point to employing more complex techniques.

### **2.7 Statistical Analysis Strategies**

Examining the general approaches adopted in statistics they are mainly seen to be technique led rather than problem led. The pattern that emerges of the technique led approach is:

- 1- Preliminary analysis
- 2- Application of technique
- 3- Diagnostic tests
- 4- Interpretation.

The application of technique, although an acceptable approach, does not provide a strategy for analysis. The approach to analysis commonly advocated is an iterative approach. Moving from data to a model until finally completing a description of

data which satisfies the analyst. Whilst this is a possible approach to find an appropriate model for a set of data, it may not satisfy a user's desire. The user may be assumed to have an objective in the study of the data and therefore this should be brought into the analysis. The approach to the analysis must be affected by the objective. In many cases it may be that the full statistical analysis is not necessary but one can stop at some intermediate point. The objective acts as a control mechanism for the analysis process.

An area which has come to light in the attempt by statisticians to build expert systems is the need for more thought about the process of analysis. The philosophical underpinning of Statistics has been well developed. Hence it is possible to characterise the reasoning processes embedded within statistical inference. However little attention has been paid to the practical problem of how an analysis should be carried out. There are some references to general approaches and even to strategies. A significant reference to such analysis is Cox and Snell (1971). The majority of comments have though focused on the application of a specific technique. Other authors have recognised this lack such as Oldford (1989) and Nelder (1989). The solution advocated by the authors is empirical, that by noting details of successful analyses then strategies would evolve. Such an approach may ultimately be successful but will require considerable effort to decide what is good and what is bad. Definition of these items might be in terms of how effectively they achieve their objective.

At a more practical level a strategy must be developed within the currently available technology. This leaves us with relatively simple tools at the present time with which to develop our system. Whilst laudable as an approach to apply a technique it does not provide a basis for analysis. There are other very general comments made about analysis. Many authors seem to subscribe to the iterative approach where either one starts from data or model then by moving between the two one finally produces a parsimonious description of the data. However few besides Cox and Snell (1971) have truly attempted to describe it beyond very vague terms.



Reliability data analysis and the software associated with it tends to fit into the linear technique based model. The analysis may well be wholly correct but wholly spurious. Any analysis must take into account the objectives it is attempting to establish. Ansell and Phillips (1989) strongly make this point adding to the iterative process the addition element of Objective, see Figure (2.2). This is not to say previous authors have not suggested its import but that it needs to be declared. Including the objective brings with a range of problems some of which are well outside the statistical analysis. It does though introduce both focus and an end point to the analysis.

Following Ansell and Phillips the aim of this research is to produce a system which encompasses the three elements 'Objective', 'Model' and 'Data'. As stated earlier the system will start with limited number of objectives understudy as well as limited models and data forms. The addition of other objectives, models or data would be straightforward and hence not to be considered a restriction.

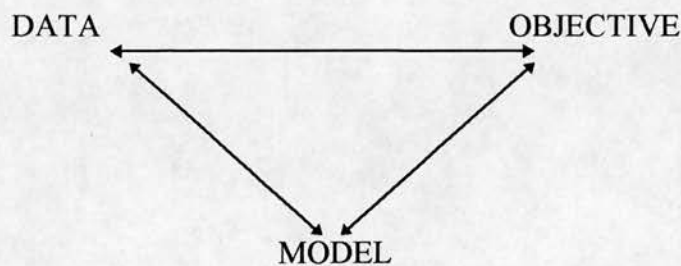


Figure 2.2 - The system's three modules.

Therefore it is desired to build an expert system that is information driven and can operate the analysis from three different approaches depending on the information available. Moving between these modules answering the user's questions. The system should adopt a non-authoritarian approaches in guiding the user through using the system and the user may diverge from the advice, and if the user does diverge then further guidance should be available on technique and analysis.



The system should be less formal than other statistical systems. This will allow the guidance in the system to be easily accessed at each stage of the analysis process. In designing the system a great emphasis is made on improving the system's user friendliness and provide the user with as much help and guidance as possible.

## **2.8 Discussion**

The aim of this research is to explore and investigate the idea of developing an expert system for statistical analysis. The area of reliability data analysis was chosen as a statistical domain to build the system since it is sufficiently self-contained and well defined. Therefore the research is involved in investigating the development of a system which will offer the user statistical guidance in reliability data analysis. Recent developments in decision support systems (DSS) and executive information systems (EIS) have had a good effect on strategic decision-making. An emphasis is made in designing this system on development of strategies clarifying objectives and into the future organising suitable strategies. The intention is to provide the user with the facility to chose and control the steps of analysis to make a decision meeting a specified objective or set of objectives.

Defining the system domain, reliability data analysis, and investigating the different types of expert systems helps us to decide on the software required to build the system and the suitable strategy to adopt in designing the system.

## **CHAPTER THREE**

### **STATISTICAL EXPERT SYSTEMS**

### **3.1 Introduction**

The improved performance of expert systems in the eighties encouraged the interest of statisticians within the field. Statisticians both have worked on expert systems for the aiding or enabling of statistical analysis and also expert systems which use statistical methods or techniques as part of their inference engines.

In the case of inference engines there has been considerable friction between statisticians and fuzzy logicians. Computer scientists had taken up the ideas of fuzzy logic to describe uncertainty. Several statisticians have argued this was a mistake and systems should primarily be based on Bayesian statistics, for example Lindley held this view, Lindley (1982). This hampered the interaction between statisticians and computer scientists as statisticians were seen as over critical of methods which appeared to work.

In this chapter several statistical expert systems will be considered. They will demonstrate the variety of strategies and approaches that have been employed. The range of applications also illustrates the diversity found. The systems have ranged from those to assist the third world in using statistical techniques to applications in the area of medicine.

Having reviewed a number of systems and ideas the chapter will progress to examine in more detail two expert systems which could be described as exemplars of their type. KENS, by Hand (1987), was designed to assist users to understand non-parametric statistics and followed a hypertext approach. GLIMPSE, by Nelder (1989) and others was designed to enable the analysis of generalised linear models.

### **3.2 General Review of Statistical Expert Systems**

There have been a number of successful systems which can be classed as expert systems. Again the main division is between the systems are those which employ

statistical inference engines to drive an expert system in a non-statistical knowledge domain and those expert systems which are aimed at assisting the user of statistics. In the first part of this section those systems using statistical inference engines will be considered and subsequently the second class will be considered.

Nelder (1990) attempted to clarify the types of expert systems in statistics. He identified 5 basic types which are:

1. Hierarchical systems which adopts diagnostic keys and the construction of trees.
2. Matching procedures - choice of design.
3. Knowledge enhancement systems which systematises statistical techniques.
4. Knowledge - enabling systems which make the techniques available.
5. Systems with higher level guidance design/analysis.

The systems are roughly in order of statistical sophistication with hierarchical systems containing fairly static information and the last system really refers to the next stage of development for expert systems in statistics.

Hierarchical systems using a tree like structure to examine the truth of a statement at each node of the tree ascending upwards to verify the top statement. Matching procedures look for the 'nearest' item in a restricted set of items. This may be in terms of data or in terms of specification. A measure of 'nearest' is required. These two are the earliest forms of expert systems in statistics.

Knowledge enhancing systems try to assist the user by producing a blue print for the analysis, aiding the understanding of the technique but not assisting the user in applying it. It may guide the user to selection of the right technique to use. Knowledge enabling systems move one step further assisting the user to select the appropriate technique amongst a range of techniques and also assisting in the implementation of the technique. These two types demonstrate the level of sophistication which has been reached in the development of expert systems.

The final type of systems should be capable of replacing the expert completely. It should be capable of assisting throughout the analysis from interpreting the requirements of the research programme into data collection to producing the final report and suggesting new lines of investigation. Unfortunately this type of system has only yet been envisaged and not implemented.

Also gazing into the future Jeffers (1989) suggests that the critical area for new development lies between the interest in actually programming expert systems and the understanding of the basis for human expertise in a particular field such as statistics. Jeffers suggests that the search for understanding is the more important at the present time, not least because it can only be addressed by statisticians themselves. The real value of expert systems is in integrating the expertise of many disciplines in the search for genuinely innovative and holistic solutions to problems. Today's environmental problems represent particular cases where holistic and innovative approaches are required.

### **3.3 Expert Systems Using Statistical Inference Engines**

Another notable early system was GLADYS (The Glasgow Dyspepsia System), Spiegelhalter and Knill-Jones (1984). GLADYS was a system using statistical methods for the inference engine. The system led to the suggestion of dealing with uncertainty in expert systems by using Good's concept of weight of evidence, Good (1950), see Spiegelhalter (1986), to combine information from different sources. This was employed successfully in combining contingency tables in which variables could not be assumed to be independent, Spiegelhalter and Knill-Jones (1986). The need for employing such was the dimensionality of the information input which required fast techniques to process the information. This was a major problem for any expert system using information collection.

The work on GLADYS may have led to development of multivariate graphs by Spiegelhalter and Lauritzen (1988). This would again be directed towards efficiently using information from a variety of sources in for example expert

systems. It promulgated the ideas of relevance and conditional independence which are helpful in reducing the complexity of graphs.

Lauritzen and Wermuth (1989) introduced graphical chain models as a class of models meant to be used for analysis of associations between continuous and discrete variables. The models are defined by first specifying the type of variables and the response structure through a dependence chain. Later conditional independencies are introduced and represented by a graph. When an association is to be investigated, the graphical structure can be used to identify how data should be used to illuminate features of the relationship. The models have a potential for turning into an expert system for the analysis of association.

Cowell and Dawid (1992) similarly addressed the problem of rapidly calculating probabilities in an expert system graph. They produced an algorithm to facilitate the simultaneous calculation of the distribution of a random variable for every node in a probabilistic expert system conditional on all the evidence obtained about the remaining nodes. Dawid (1992) showed how a wide variety of computational tasks relating to a probabilistic expert system (PES) may be solved by the variant of a general propagation algorithm, operating by the passage of flows between neighbouring cliques (representing suitable subsets of the variables) in a high-level junction-tree representation of the system.

An alternative Bayesian view of expert systems was taken by Goldstien (1989) with the B/D system. (B/D is an acronym for belief adjusted by data.) In the B/D system a subjective approach is taken based on exception which requires only a partial specification from the user. A minimal rationality criterion is suggested to establish temporal coherence. The system demonstrates how beliefs may be represented, manipulated and explained by belief transforms. The system also enables the user to revise a belief by representing and criticising it using the facility of data trajectory.



The work of Spiegelhalter, Lauritzen, Dawid, Goldstien and others were primarily directed towards expert systems employing statistical inference engines. Whilst some of the ideas and concepts have parallels with the current system they are seen to be tangential to the main thrust of the current research.

### **3.4 Expert Systems Assisting or Enabling Statistics.**

An early system was developed by Barrie Whetherill for the Overseas Development Agency systems for regression and surveys, U-REG and U-SP. These systems enabled the user to employ statistical techniques whilst providing assistance in their use especially with the diagnostics. The systems were built in APL.

There are a variety of MYCIN-LIKE systems which are rule based diagnostic system with a tree shaped rule-base and truth functional (compositional) propagation of uncertainty. As Hajek (1989) points out such systems have been subjected to serious theoretical criticism and seem not to find theoretical interest any more, but their techniques survive in commercial shells, like CRYSTAL see chapter four.

As THESEUS, Bell (1989), is an illustration of a rule based system with expertise in the area of completely randomised designs and multiple comparisons. Its major contributions in the area of providing the user with appropriate explanation facilities. The provision of some form of explanation facility is usually regarded as a desirable attribute of expert systems but practice has proved it to be a difficult task. In most expert systems this facility took the shape of tracing of rules which normally did not really help the user to understand the reasoning and strategy behind the system. In THESEUS help is available based on key words and trace arrays are kept of rules and goals. There is also a basic WHY facility which informs the user of the rule currently being tried and, where relevant, what the current goals are in the system. It includes other facilities such as (strategy text) which is to provide guidance on the strategy used in the system separate from



semantic help. The 'WHY?' facility lists the current goals and rules and allows the user to look at other rules tried. 'WHAT...IF?' facility enables the system to work through the rules explaining as it goes along, until it reaches a point where more information is needed. Goal-snapshots provide a pictorial representation of the current and past goals in the system. Finally 'EXPLAIN' gives a pictorial way of moving through the decision made by the system and which will highlight the critical parts of rules used.

An example of a knowledge enhancement system is Linear Modeling Guide (LMG) by Lunneborg (1990). It is a system to assist data analysis in the behavioral and biological sciences. It provides the analysis of explanatory models for empirical data. It is organised as a collection of screens each consisting of a card of hypertext to be displayed and a macro set of instructions to be executed. It uses hypertext extensively not only to enhance the user's knowledge of linear modeling but to explain its results and engage the user interactively in making modeling decisions.

LMG's knowledge base and action rules are packaged as discreet, limited screens to make the system easy to maintain and extend. The user's responses to calculations on the data and to hypertext cards have a major influence on the presentation of subsequent hypertext. In this way LMG adjusts itself to the user's knowledge, providing only as much guidance or explanation as is requested in making successive modeling decisions and intervening when it senses a difficulty of which the user may be unaware. LMG operates within an MS-DOS statistical package, PC-ISP.

Another interesting system conceptually is ESTES by Hietala (1988). The system was intended to provide guidance for an inexperienced time series analyst in the preliminary analysis of time series, i.e. in detecting and handling of seasonality, trend, outliers, level shifts and other essential properties of time series. ESTES is designed to exploit the users knowledge and experience of time series being constructed. Even the case of an inexperienced user he/she may have plenty of

useful knowledge concerning the environment of the problem in question. ESTES can detect any conflict if it exists between the initial results computed by the system and the knowledge illicited from the user. And, if necessary the system also carries out more extensive analysis and apply more sophisticated statistical methods. This is aimed to minimise the number of unnecessary reasoning and calculation steps. ESTES was implemented on Apple Macintosh personal microcomputers using a combination of PROLOG and PASCAL languages.

Oldford (1989) suggested that as statistical analysis becomes more automated it will be possible to build up routines for analysis of data. These routines will form the basis of a strategy. Repeat analysis will develop the analysis in a learning process. An expert will have to be involved with the repeated analysis so that decision can be made about the quality of analysis. To illustrate this approach Oldford developed DINDE, Oldford (1988).

DINDE network model is used to build new statistical programmes by interacting directly with the network display of the analysis. In this way patterns in analysis can be captured and applied in new situations. He refers to this style of programming as graphical programming. The system will provide an integrated programming environments and has been referred to as experimental, or exploratory, programming, see Sheil (1984). The premis is that a great deal of systems programming involves making small incremental and often exploratory changes to the software. To support this activity, many programming tools such as structured editors, source level debuggers, incremental compilers, data structure inspectors, and various performance monitors and meters, are integrated into a single programming environment.

Oldford suggests that a statistical analysis usually proceeds in a tentative fashion, alternately entertaining and discarding models which address particular aspects of the substantive under study. A good analysis is typically developed in a gradual fashion, as opposed to being first prescribed and then implemented. Indeed, the

outcome of the analysis process is often a more clearly defined problem than a concise solution.

This view of statistical analysis has a number of implications for the design of a statistical analysis environment. In particular, an integrated programming environment is regarded as a natural base for a statistical analysis environment, see McDonald and Pedersen (1988). This environment is further enriched by adding a great deal of information about statistics in the form of new data structures, procedures, and graphical interfaces.

Oldford suggests five areas of concentration that are being explored at present to develop these ideas. The areas are,

1. Numerical methods (easy access for the environment)
2. Statistical programming support
3. Data representation and support
4. Integrated interactive statistical graphics
5. Analysis management

In a similar fashion Naeve suggested that statisticians should strive for statistical environments, workbenches for statisticians, that among other things should have AI features such as rules, frames inference strategies (i.e. backward chaining, forward chaining), see Naeve (1989). Naeve also suggested that by breaking statistical expertise into small 'special' purpose expertise statistical analysis would be considerably easier. Meyer suggested that the development of an expert system for the design of experiments requires knowledge of high-level problem solving strategies used by expert statisticians in the consultation process.

Naeve feels that artificial intelligence concepts and tools should be exploited rather than trying to build more or less stand-alone statistical expert systems. This would assist in developing the statistical environments (workbenches for statisticians). Naeve suggests that these ideas can be implemented in S. Inference facilities were

built using a commercial shell (ZENO) which produces modifiable C code. A prototype system called GUARD has been implemented. In a later paper, Neave et al, see APL conference Toronto (1993), develops the idea of moving from statement of analysis to analysis using a development of the GUARD system.

### **3.5 Other Statistical Expert Systems**

Obviously within Statistics there are generally two types of Expert Systems The first group are those which use Statistics, and particularly statistical inference for example GLADYS (etc). The second group which we will focus on, is these which assist statistical analysis. This group consists of systems such as KENS, Hand (1987) and GLIMPSE, Nelder (1989).

Considering those systems which aid statistical analysis that have already evolved they are either Knowledge Enhancing such as KENS, or Knowledge Enabling such as GLIMPSE. Knowledge Enhancing can be simply regarded as computerised text supplying the user with required knowledge. An example would be on how many observations you need to reach a specific power of a statistical test. Knowledge enabling not only supply the knowledge but also the ability to carry out the text. Given some data with a description the system may suggest an analysis then carry it out. Glimpse within the narrow confines of Generalised Linear Models will advice users on the next step in an analysis. It is of course a technique based approach since it focuses on the technique of generalised linear models and not the user's objective for the whole analysis.

#### **3.5.1 KENS, ( A Statistical Knowledge Enhancement System )**

KENS is designed to have general applicability, within the domain of non-parametric statistics. It does not seek to solve problems for its user, but to assist the user to solve problems and to improve the user's understanding of non-parametric statistics. Hand diagnosed several problems in existing expert systems which influenced the development of KENS. These are:

1- The rule-based structure of most current expert systems is suitable to tackle diagnostic type problems where the aim is to choose one of several possible answers. This rule-based architecture is less suited to cases where the objective is not clearly defined, or when the user is faced with a large number of different types of questions. Also rule-based systems are more suitable for well structured artificial domains such as configuring computer systems, while their success is limited in more natural, less well structured, domains.

2- The lack of clarity in the presentation of solutions by classic expert systems. The lack of structure of user/system interaction and the user finishing with more information than was relevant are features of the performance of rule-based systems.

3- The information explosion and the increased demand for locating and updating material.

4- The need to be reminded of information not used regularly.

5- The failure of text books and manuals to represent knowledge adequately.

The basic structure of KENS is a core consisting of three graphs (semantic networks), these are:

1- The Concept Graph.

2- The Relationship Graph.

3- The Reference Graph.

A graph can consists of several sets of nodes. The first set of nodes contains small collections of knowledge called 'frames'. The second set of nodes are labelled by words or phrases named 'descriptors' which hold description of some aspects of the first set to which they are connected by the edges of the graph. Each frame is

described by the descriptors linked to it. The links between the frame nodes and the descriptor nodes come in a number of different types called 'qualifiers' such as the Null, Definition and Example Qualifier.

Although KENS has been successful in achieving the goals it set out to achieve as a knowledge enhancement system it is limited and can be described as a text book on a machine. Unlike most other statistical expert systems KENS assumed little knowledge of statistics by the user.

There were exceptions to this rule like in the GLIMPSE system, being developed by Nelder and colleagues e.g. Wolstenholme and Nelder (1986). This system requires the user to have a certain degree of knowledge about Statistics. It helps the user to produce better analysis.

### **3.5.2 GLIMPSE, Generalised Linear Interactive Modelling with PROLOG and Statistical Expertise**

GLIMPSE is a knowledge-based front end system which provides a task language as part of a high-level interface to GLIM 3.77. It arose out of an Alvey funded project to investigate the use of logic programming techniques and tools to develop front-ends to large software packages such as GLIM. The aim of GLIMPSE is to provide assistance and optional guidance in the application of generalised linear modelling. The front-end uses the logic-based expert system shell APES which is written in LPA Sigma-PROLOG. APES provides a declarative dialogue in which the user can supply statements which are checked against the relationship within the rule base. It yields explanation facilities following the computational steps in an analysis. There are three main parts of GLIMPSE; the "translator", the "abstract statistician" and GLIM which is the standard GLIM 3.77 program. The translator translates GLIMPSE commands to GLIM commands. It provides a richer medium than standard GLIM. No longer is there a need to either proceed through tedious declaration of variables or use so many macros. Hence a plot may be specified for a transformed variable without transforming the variable. Obviously this is an advance on GLIM, but one questions why it was not already incorporated in the GLIM package. There is no



real need for the PROLOG front-end to deliver this facility. Yet since it is within the present structure one pays the penalties of having to be more confined . Data entry, at least on the initial attempt, can be both tedious and hazardous.

### **3.5.2.1 The Structure Of GLIMPSE**

The structure of GLIMPSE consists of three main components :

1- Abstract Statistician. This contains the expertise about statistical analysis and the use of GLIM 3.77. The abstract statistician is portrayed as an eye overlooking the analysis. It contains the distillation of advice/expertise from senior statisticians. At any stage one can ask it for advice on what to do next. Advice is supplied and this can then be questioned. This is the central plank of the system, since it is the expert part of the system.

2- Translator. This translates GLIMPSE's command language into executable GLIM commands.

3- Statistics Package. This is GLIM 3.77.

The structure for GLIMPSE modelling can be described as "directed graph structure" , the number of nodes linked by directed lines representing the stages or activities involved in the statistical analysis of data when using generalised linear models. These activities are :-

1- DI (data input). This involves storing and accessing of numerical values. Each of these values is given its own reference name.

2- DD (data definition). This involves gathering the information available about the "response and explanatory variables", the type and structure of the measurement, whether experimental or observational, and the kind of analysis needed.



3- DV (data validation). This involves the detection of errors and inconsistencies in the numerical values of response and explanatory variables.

4- DE (data exploration and transformation). This involves the possibility of transforming either the response variable or the explanatory variables.

5- MS (model selection and specification). This involves the modelling of a particular response random variable by one or more explanatory variables.

6- MC (model checking and assessment). This involves checking the suitability of models selected by :

a- tests of deviations in particular directions.

b- visual displays.

c- the detection of influential points.

### **3.5.2.2 Using GLIMPSE**

A statistical analysis with GLIMPSE involves the user supplying the system with data and then performing various actions on that data set. APES provides a declarative dialogue in which the user can supply statements which are checked against the relationship within the rule base. It yields explanation facilities following the computational steps in an analysis. There are many criticisms about the system, some which may have been overcome through familiarity with it. The cost to the end user is high to gain familiarity.

## **3.6 Discussion**

The aim of this chapter has been to review statistical expert systems and the main developments within such. There is obviously a wide variety of statistical expert systems and a diversity of applications. Statistical expert systems divide into two groups those which employ statistics as the inference engine and those which assist

statistical analysis. In the current research the emphasis is on the latter, though of course ideas and concepts from the former may provide assistance in achieving the desired system.

Of the systems designed to assist statistical analysis Nelder suggests there are five types. Of these five systems it would appear that the first which might be described as the MYCIN-LIKE approach and the knowledge enhancing and knowledge enabling systems are those which could be explored within the current research project. Chapter four will investigate using a MYCIN-LIKE system called CRYSTAL for the current research project. Of the other two types of system it would appear that the knowledge enabling systems would seem a better way forward within the current research. Hence it is the GLIMPSE blue print which would seem to offer more in the current project rather than the KENS approach.

GLIMPSE statistical advisor seems a concept which would be usefully employed in the current system. However it would also seem that the ideas and concepts suggested by Neave of a statistical environment could also be contemplated as a model for the current project. In some senses the work described in chapter six and seven would seem to be an attempt to marry some of the ideas from Nelder and Neave within a system.

## **CHAPTER FOUR**

### **PHILOSOPHY OF DESIGN**

## **4.1 Introduction**

In some respects, the process of developing an expert system is similar in its stages to those involving the development of a conventional system. The first task would be to identify problem area and its knowledge domain. Having identified the problem and decided on the source of expertise, the second stage in system design is to detail the features needed to be incorporated in the system and the selection of suitable software. Obviously the selection of software is influenced by system's features and also by the design philosophy.

One of the important tasks facing an expert system developer is the need to consider the mechanism which will provide the underlying reasoning process. In chapter three a wide variety of statistical expert systems were reviewed. This diversity did not suggest a particular approach, though, of course the knowledge enabling system GLIMPSE seemed an appropriate "blue print". Also the concepts of Olford (1989) and Neave (1989) seemed attractive. It was decided to explore a range of alternatives to such systems and different approaches to implementation such as hierarchical systems, frame based systems, using expert systems shells and expert systems languages.

In this chapter the intention is to discuss these various approaches of design that were contemplated at the early stages of this research and analyse the positive and negative features of these design approaches.

## **4.2 Hierarchical Systems**

A hierarchical design will consist of nodes that represent each stage of the statistical analysis. These nodes are linked in a hierarchy. The linking is usually through rules in an expert systems. An attempt to follow this approach in designing the system was made using the software package CRYSTAL. CRYSTAL (1986) is an expert system builder and can be considered a MYCIN-LIKE system. It gained a measure of acceptance having been implemented by a number of firms. It

is a fairly user friendly system though simplistic in structure. The system is rule based with a hierarchical structure which assumes a tree like devolution of knowledge. It is a simple structure and fits well into computational framework by making sequential movements through the decision tree.

The package CRYSTAL is used to build the expert system with an interface with the spreadsheet LOTUS 123 to input and manipulate data. An emphasis is made on giving the user the facility to choose and control the steps of analysis without having to follow a set routine of operations and processes.

#### **4.2.1 The Structure of The System**

The structure of the system consists of three main components, the expert system which includes a statistical advisor that contains the expertise about statistical analysis, the front-end of the system and the knowledge-base, and finally a set of statistical Reliability models. The knowledge base of the system consisted of a set of rules. The rules are very simple being "horns" clauses with the structure, see figure [4.1] and example of such is given in figure (4.2).

**CONDITION IF CONDITION  
AND  
OR CONDITION**

Figure 4.1 - The structure of rules in CRYSTAL

CRYSTAL has a number of features including the Retest possibility and interfaces to other software such as databases and spreadsheets. In the latter case of spreadsheets the interface did not on investigation seem very active but mainly used to report material from or to a spreadsheet. It is possible with considerable effort to design relatively sophisticated features into CRYSTAL including the possibility of frames. However these do require considerable distortion of the software reasoning.

## MODEL

```
IF      CURRENT STATE

      AND  DO: TEST EXPRESSION
            DOC$="NO"
      AND  DI: ASSIGN VARIABLE
            MODEL$="UNKNOWN"

OR      DO: TEST EXPRESSION
            TRENDS$="YES"

      AND  DO: ASSIGN VARIABLE
            MODEL$="NHPP"

OR      DO: TEST EXPRESSION
            DEP$="YES"

      AND  DO: ASSIGN VARIABLE
            MODEL$="BPP"

OR      DO: TEST EXPRESSION
            HAZ$="YES"

      AND  DO: ASSIGN VARIABLE
            MODEL$="HPP"

OR      DO: ASSIGN VARIABLE
            MODEL$="OTHER MODEL"
```

Figure 4.2 - A typical Crystal rule.

The hierarchical structure has been retained and each stage of the statistical analysis is represented by a node. These nodes connect the different activities involved in the analysis, see figure (4.3).

### **4.2.2 Activities of The System**

There are several stages and activities involved in the system, a description of the significant features are :



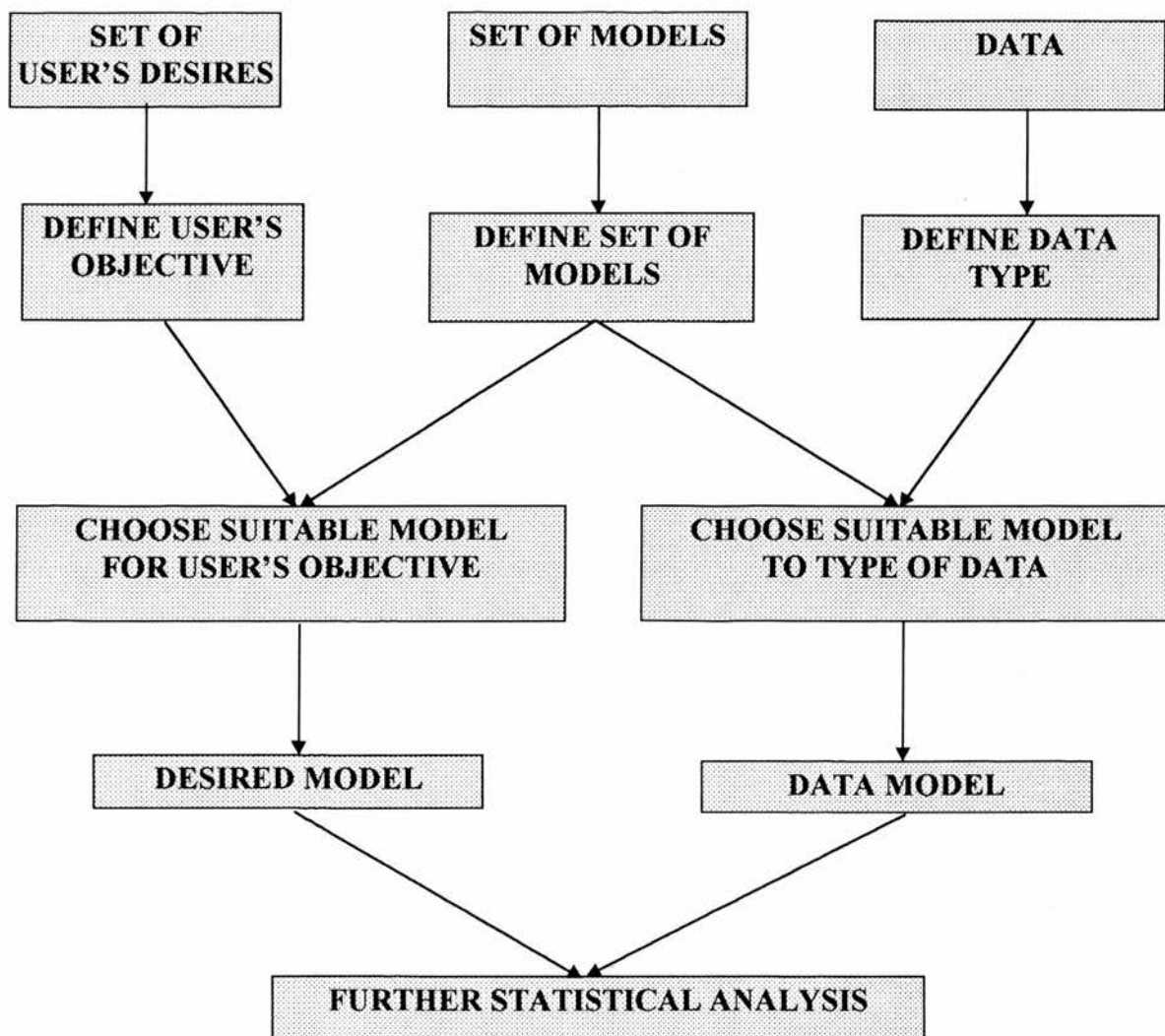


Figure 4.3 - The hierarchical structure of the system.

### 1- Data Definition

This involves defining the type of data. CRYSTAL allows definition of the type of data. The data may be single values and single or double dimension arrays. The function dealing with data definition will interrogate the user as to whether the data is a single value of single or double dimension array. It will ask for the name of data and the number of values in the string of single values or the array. It is possible to input values of an array and update the position number (1st, 2nd...) of

the values being entered. It is also possible to display parts of the data. The display function will ask whether the user wants to display a single value, a set of values or an array. The display function can display a string of values of up to four values, any more than that an error message appear for the string being too long to concatenate. It was hoped to resolve this problem by using the interface to LOTUS 123. As will be seen under Data Input this was not successful.

## **2- Data Validation**

This involves error detection of data. This is achieved by checking the dimension of the arrays of data used and checks the range of values. Obviously no lifetime can be negative.

## **3- Data Input**

This involves storing and accessing data. To input data proved to be more problematic than expected. In the early stages of the research an attempt was made to use CRYSTAL only to input data. The main problem with such an approach was handling arrays and not being able to display the whole of the array on screen. Then several attempts were made to use the interface with LOTUS 123. The attempt to export data from CRYSTAL to a worksheet in LOTUS faced several problems: not being able to input values in desired positions in the worksheet, and also the slowness of the process meant a relatively longer time is needed to input data. These problem were solved by transferring data to an already opened worksheet with 0's in all possible positions of values, but this meant that the data file in LOTUS has to be created externally and not from CRYSTAL. These features of the interface led to the abandonment of CRYSTAL eventually.

## **4- Data Model Selection**

This involves choosing the appropriate model for data. This was achieved by following a decision tree diagram of models by Mcdonald and Richards. Using

CRYSTAL it is possible to test for the different distributions presented in the diagram and decide on the appropriate model. An example of a set of rules to acquire and manipulate data is given in appendix (4).

#### **4.2.3 General Discussion**

Attempts to build an expert system were partially successful in the design of data entry and model relationship. A system in CRYSTAL can have a modular structure, however the hierarchical structure of the system does restrict the user to a predetermined set of paths of analysis. This made the system authoritarian and not as flexible as desired. To achieve the desired level of flexibility a considerable amount of development work was required. The lack of substantial numerical routines available in CRYSTAL and the lack of Graphics were drawbacks to using CRYSTAL. The problem of slow processing caused by the use of variables to hold previous knowledge did not make CRYSTAL a good system to develop the system through. CRYSTAL can be useful in dealing with the design of simple tasks such as data entry, but it can be very restrictive when we attempt to explore the analysis process in the system. Generally the comments on CRYSTAL are the same as any MYCINE-LIKE system, see Hajek (1989). See appedix 3 for full knowledge base listing.

#### **4.3 Frames**

A frame is a collection of information about a context. Frames were proposed by Minsky as a unified knowledge representation environment. Each frame is a distinct object combining both data and procedures. Frames are linked together into frame systems, and such systems have been applied to a number of diverse application in Artificial Intelligence. The information is referred to as attributes of the context. Frame systems are made of a set of frames linked together. These systems have been successfully adopted in other areas of Artificial Intelligence such as natural language, understanding and scheduling. PROTEUS (Russinoff 1985, and Petrie 1987, Poltrock 1986) is an example of such a systems. It is a hybrid expert system written in COMMON LISP, developed at the Micro-

electronics and Computer Technology Corporation, see Kim, Fredrick and Lochovsky (1989).

In some frames systems efficient methods and procedures are used to combine data and to represent knowledge. One such approach is to have:

- 1- A name which identifies the concept it describes.
- 2- A description which consists of a set of slots describing the various elements of the concept dealt with.
- 3- Spaces in a frame, next to the slots which may be filled with information representing the value of the slot , see Tello (1988).

<b>NAME :</b>	<b>SM-1</b>
	<b>Specialisation of : Student</b>
	<b>Name : H. Brown</b>
	<b>Age : 32</b>
	<b>Address : 19/11 Bristo Place, Edinburgh</b>
	<b>Department : Business Studies</b>

Figure 4.4 - A typical Frame

To connect frames in the system pointers are used which are in frame slots. These pointers are the frame names of parent/child frames. Frames are organised into complex hierarchies and a child frame can inherit both the data and the attached procedures of its parents. Multiple inheritance is an explicit feature of frame systems and give these systems a lot of flexibility and power. The ROOT in a system contains the most general procedures and values that are inherited by every other frame in the hierarchy.

The inheritance mechanism is implemented by using pointers divided into two types of slots. AKO (A-KIND-OF) slots point from the child frame to its parents. Parent frames contain INSTANCE slots which point to child frames. It is usual to have a number of pointers in both these types of slots, see figure 4.4.

A child slot may inherit values and procedures from a number of more general parent slots, (figure 4.5). A parent slot is likely to have a number of INSTANCES of child slots.

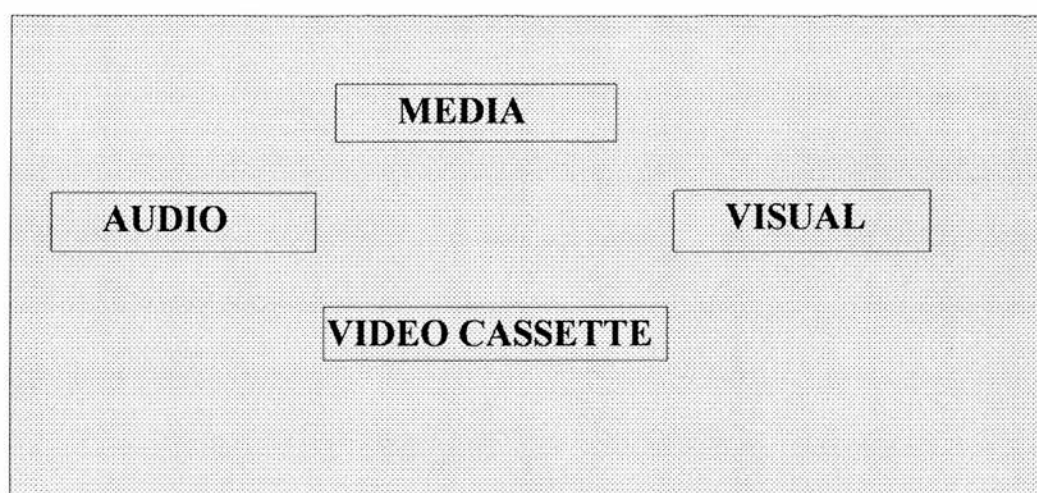


Figure 4.5 - Multiple Inheritance.

#### **4.4 Frames and Objects**

There are many similarities between frame systems and object oriented systems (OOS). In many cases there are one to one mapping between concepts, see figure (4.6). One of the main differences between OOS and frames is in the method used to invoke procedures and create new objects. OOS uses a method passing system, each object recognises its own set of messages and acts according to the message received. Therefore many procedures can have the same name, it is the type of message that determines which procedure is invoked at run-time.

Frames do not support message passing as such, though some workers have added explicit message passing mechanisms to frame based systems. However it is arguable that the inheritance path through AKO links, provide an implicit message passing mechanism.



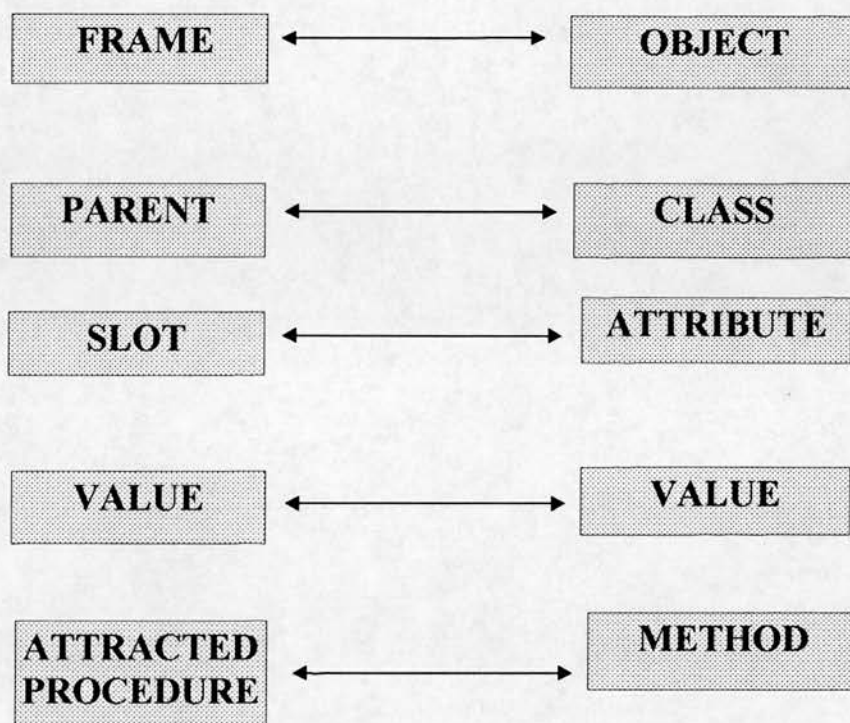


Figure 4.6 - The mapping between frames and objects.

It is advisable to have the most general procedures attached to the frames nearest to the root, and to have more specialised procedures distributed where necessary in the system.

If a frame requires to be deleted it is not deleted directly. A frame has a distinct position in the hierarchy and the destruction of a parent frame may leave a number of orphans in the system, which are left floating. Therefore destroying a frame can only be done by the parent and any children of the destroyed may also be deleted.

Multiple Inheritance which is the main feature of frames systems, is the capability of a child frame to inherit the procedures and data of its parent frames. System's that adopt this approach have an improved flexibility to that hierarchical systems such as the ones built using CRYSTAL.



However the final development of the system can be demanding on time and effort because of the number of frames needed to build the expert system and incorporate all the required knowledge. Also an extensive amount of work is required to achieve needed interaction between frames and the vehicle to do the numerical calculation.

#### **4.5 PROLOG Based Systems**

PROLOG (which stands for programming in logic) is a computer language based on predicate logic, in particular the clausal form of logic and resolution inference. It was first implemented in 1972 in Marseilles by Colmerauer and Roussel as an interpreter in the medium level programming language ALGOL-W, see Colmerauer (1973) & Roussel (1975). Several subsequent implementations followed including Edinburgh's compiled version DEC-10. Interest in PROLOG has been stimulated by the Japanese decision to use it as the core language for which they will design their fifth generation computers.

PROLOG is an attempt to implement predicate logic, hence the language is organised to establish truth or falsehood of statements. It is a list processing language. Knowledge is represented in a number of objects the main one is a word. These words allow the construction of other objects in the form of a hierarchy of lists and sublists. Relations in PROLOG can be alpha-numeric constants. Their forms depend on the types of relations as described in the Table (4.1).

<b><u>Type of Relation</u></b>	<b><u>Relation</u></b>	<b><u>Notation</u></b>
Unary	Type of Incident	Infix
Binary	A greater-than B	Infix
Ternary	Takes (event 1, event 2, UK)	Postfix

Table 4.1 - The types and notations of relations in PROLOG.

Setting up a database is performed by typing in the relations and facts about the objects involved following some standard rules. The key words used in PROLOG to manipulate the database are ADD, ACCEPT, LIST, DELETE, EDIT and KILL. Such utilities are helpful in speeding the development of a database.

It was apparent that for a realistic system a large number of rules would be needed. Care would need to be taken in developing the system because the order of addition of rules is also the order of search. Also PROLOG is not a natural language, but suffers from many peculiarities, and is generally user unfriendly. So to make it acceptable we would have needed to build a number of sophisticated utilities. Utilities to improve its userfriendliness. PROLOG's controls processes are obtained through, Backward Chaining and Backward Tracking. In the version used, Micro-PROLOG, it was difficult to construct a forward chaining procedure since it is primarily designed for Back Chaining. All these problems make the process of using PROLOG in building expert systems a time consuming process.

Obviously in a deeper analysis one could further investigate the power of PROLOG for example its declarative power while constructing meta forms.

#### **4.6 Discussion**

During system design and evaluation, development of optimal design approach for an intelligent system can be a complex process. There are many tools and a large literature that combines the expert system and statistical modelling paradigms. The selection process of an appropriate approach of design to the expert system must include consideration to the main features of the expert system. The system needs to be non-authoritarian allowing the user the freedom to decide on the analysis path without going through a rigid sequence of menus of operations and processes. The system requires to be easily expandable and capable of catering for possible strategies of analysis which may be adopted during analysis. Therefore a hierarchical structure is not the ideal choice since it will compel the user to follow a fixed set of possible analysis paths. The system is expected to provide

knowledge and carry out statistical techniques and methods. This rules out knowledge enhancing systems since they do not provide the facility to carry out processes and restrict output to hypertext like knowledge.

Taking these points into account also the limitation on time and resources it was felt that the system needs to be a non hierarchical knowledge enabling system. An object oriented design approach is preferable to a frame based approach since the latter is time demanding due to the large number of frames needed to incorporate knowledge and develop interaction between system's modules. An object oriented design offers the capability to develop the system and retain its desired features. This will be discussed in more detail in chapter five.

Following this a great deal of synthesis work remain to be done to develop the appropriate design for the desired expert system. This will be discussed in greater detail in chapter six.

## **CHAPTER FIVE**

### **OBJECT -ORIENTED PROGRAMMING**

## **5.1 Introduction**

Computer scientists, both programmers and software designer, strive to improve the quality of software. They primarily achieve this by producing philosophies of programming or design which will enable the production of reliable software in efficient manner. These usually are specified in terms of principles of design or strategies of production. There exists a wide range of possible principles and strategies to choose from. An examples is structured programming, for example Jackson Structure Programming, see Jackson (1975).

As was indicated in the discussion of the survey it was not envisaged that the design of the system would be achieved by refinement of an initial specification to implementable code. Therefore, at least at the meta level, it was decided not to use the design approaches which employ this technique. The prime reason for this course of action was to retain flexibility and the ability to innovate so that an exploratory approach could be adapted. It was therefore felt that a modular approach would seem most sensible way forward. Object oriented programming which offers modularity therefore appeared to be an appropriate approach to tackle the current project. Object oriented programming provides many advantages without imposing a too restrictive framework and these will be discussed later in the chapter.

The aims of this chapter are two fold: to characterise object oriented principles in relation to the specific project and secondly to review object oriented languages or languages which could be adapted to object oriented principles. In characterising object oriented principles there is a need for a clear definition of what constitutes an object oriented language. This is far from easy since the principles have developed from programming languages and subsequently an external coherency was imposed. A rationalisation for the underpinning philosophy of design has been based on ontological justification. The definition of an object oriented system would seem to rest on a set of specific features of the system centred around the concept of an object. These features will be discussed in the next section.

Object oriented programming developed from a series of programming languages. Hence there exists a number of possible object oriented languages which could have been used to build the envisaged system. Not unsurprisingly these languages do not conform to all the object oriented principles. Some of the major object oriented languages will be explored: CLOS, C++ and SMALLTALK. Unfortunately these languages did not provide the desired features for the envisaged systems. It was therefore necessary to explore alternative languages to see if they could offer the desired features and yet allow object oriented principles to be used. It was decided therefore to explore whether APL2 could conform to object oriented principles since it offered features which were desirable in other respects.

## **5.2 Definitions**

Object oriented programming developed from implemented languages. The principles of object oriented programming, therefore, are a distillation of the features of these languages and its definition can be regarded as a search for commonality amongst the languages. However, the concepts have been refined so that general concepts have emerged which do not necessarily reflect a specific implementation or a particular language. Therefore it could be suggested no language satisfies all the object oriented principles.

The main idea of the object oriented paradigm of programming is the view that a program consists of independent objects that communicate via messages, see figure (5.1). Whilst this is an elegant view of programming it does not immediately yield a full specification of the principles. The detailed specification arose from the implementation languages. Unfortunately this has led to some confusion about the paradigm. As Kim and Lochovsky (1989) suggest less confusion would have arisen if the principles had emerged from modelling consideration rather than from implementation.

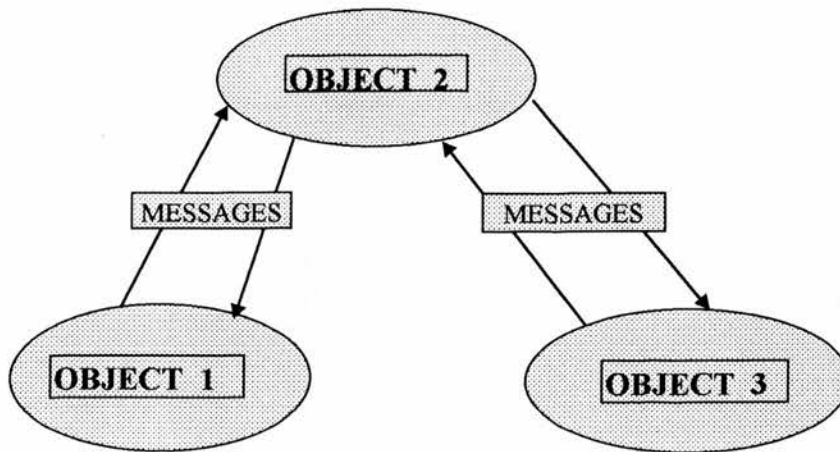


Figure 5.1 - Communication between objects

Since the detailed specification is derived from implementation, the variety of implementations have given rise to a range of possible concepts for object oriented programming. This diversity has been increased as implementors have described their own system as object oriented often with little justification. It has also been the case that implementors have defined the concepts in terms of the features of their own system which are either felt to be noteworthy (interesting) or useful. Therefore object oriented programming can be interpreted in a multitude of ways.

There are fortunately certain facets which can be brought together to form the basis for a definition. Object oriented computing is a style of computing in which data and associated procedures are encapsulated to form an object. An object is a useful computing entity existing at a higher level than procedures or data structures.

The other key features of object oriented programming are:

- 1) Abstraction
- 2) Encapsulation
- 3) Independence and persistence
- 4) Homogeneity
- 5) Inheritance
- 6) Polymorphism.



Each of these features will be explored in the subsequent sections.

### **5.2.1 The Object**

An object within the paradigm is a 'package' which contains the data and the procedures/operations which are normally performed on the data. Objects provide a courser level of granularity for program decomposition than is available by using data or procedures. It would appear a more natural decomposition than using data or procedures. For examples stacks, queues, file interfaces, sensor interfaces, robot interfaces, compilers and operating systems can be considered to be objects.

As figure (5.1) illustrates these objects interact through messages. For example object 1 in figure (5.1) could be the main programme and object 2 could be printer driver and object 3 could be the printer. A message is sent from the main programme to the print driver which then passes a message to the printer. Once the job is completed then a message is sent back to the main programme.

### **5.2.2 Abstraction**

A central facet of the object oriented paradigm is the user only needs to know what a module does and not how it achieves the task. As Nierstrasz states 'By far the most important concept in the object-oriented approach is data abstraction. By this we mean that we are interested in the behaviour of an object rather than its representation'. Therefore the designer once an objects has been created with its capabilities only needs to know how it behaves.

Since objects communicate by messages then an object's behaviour is defined in terms of its responses to the messages it receives. Once an object is created the designer is only interested in the responses to specific messages the object receives. A message causes the object to behave invoking a method of the object. The methods can be regarded as a definitions to responses to possible messages, Banerjee et al (1987).

Another aspect of abstraction in object oriented approach is that those features which do not need to be shown are hidden. Hence the procedures and data of an object are not revealed unless it becomes necessary to do so. This can be seen as a form of protection for the data and procedures of the object. In a fully object oriented language these items would be wholly protected allowing no interference from outside the object.

### **5.2.3 Encapsulation**

The behaviour of an object is encapsulated in its methods. Methods are mechanisms which have access to and can change the 'state' of the object. Thus an object is described in terms of form of its instances (instance variable) and the operation (methods) applicable to its instance variables. The instance variables, together with the methods, are called the properties of the object, Banerjee et al (1987).

From an ontological viewpoint things must change and changes cannot be described separately from things. In this respect, encapsulation is a fundamental principle. However, the notion of methods or any other mechanism for changing the state of an object does not exist. Instead the concept of a law defines the lawful or 'allowed' state (i.e. combination of state variables). Laws are viewed as properties of things, see Meyer (1988).

### **5.2.4 Independence and persistence**

The idea of independence incorporates two characteristics of an object. One is related to the state of an object and the other to its existence. This is expressed by Nierstrasz (1987) as 'Objects also have control over their own state. Once created, an object will continue to exist ( or persist ) even if its creator dies '.

Independence implies that the only way an object can change its state is through the action of its own methods, namely, through its dynamic characteristics. Therefore encapsulation is a necessary condition for independence. In ontology,

the possible states of a thing are decided by the laws of the thing. These laws are considered properties of the thing.

Persistence of a thing manifests through the principle of 'nominal invariance' that enables changes of the state without changing the essence of the thing. There are no explicit mechanisms for the creation or destruction of things. Having been created an object will continue to exist without necessary reference to its creator.

### **5.2.5 Homogeneity**

Homogeneity implies that everything is an object. In particular, for complete homogeneity, messages and properties (instance variables) should be viewed as objects. This approach obviously leads to circularity, as indicated by Nierstrasz (1987), because if messages are viewed as objects, then they should send messages to communicate with objects, and so on. Similarly, if attributes of an object are objects, they should have their own attributes, which are objects, etc. Therefore one has to stop arbitrarily at a certain level in order to break the circularity and the ambiguity which arises. The relevance of this for the system developed within the project will be discussed later.

Ontology makes a clear distinction between things and their properties. In this case, there is no complete homogeneity. Moreover, properties can not exist independent of things. Therefore there is asymmetry in the fundamental characteristics of the two concepts. Ontology allows for things to be composed of other things, and also makes a specific assumption that there exist simple things that can not be further decomposed. Therefore the existence of a basic level as a principle is asserted, and is not an arbitrary decision. Also ontology adds the distinction between resultant and emergent properties, thus formalising the idea that a thing may be totally different from the collection of things in its composition. This view implies that state variables of a thing belong either to its composing objects or only to the thing itself, see Ayer (1986).

### **5.2.6 Inheritance**

This provides the mechanism for managing objects in ways which enhances their reusability. Inheritance refers to the ability of one object (the child) to inherit some of its behaviour from another (the parent). If a child can inherit from more than one parent then this is said to be Multiple Inheritance, see Meyer (1988).

Objects can be grouped together into classes. A class is a definition of an object type. All objects in a class have the same instance variables and methods and respond to the same messages. 'A class describes the form (instance variables) of its instances and the operations (methods) applicable to its instances', Banerjee et al (1987). Objects can be categorised into subtypes or subclasses through specialisation. Objects in a subclass inherit all the instance variables and the methods from the superclass, but may have additional instance variables and methods. In object oriented literature class hierarchy is viewed as a mechanism to provide for reusability, savings of information in databases and programming, and simplicity, Banerjee et al (1987).

In ontology, classes are defined based on the idea of the scope of properties. The concept of a class is extended to a kind by considering the combined scope of a set of properties, and further to a natural kind by considering the scope of a set of laws. Therefore, both 'static' properties (attributes) and dynamic properties (laws) are used in the definition of kind hierarchies. It can be proved that the collection of kinds forms a complete lattice under inclusion, Bunge (1977). Hence ontology's view of classes equals the idea of classes in the object oriented approach.

### **5.2.7 Polymorphism.**

The ability of an object to take more than one form. This can be manifested in two ways. Firstly the same name may mean different things depending upon the type of object being referenced. Secondly for strongly typed languages polymorphism

represents a relaxation of type checking. So if a function calls for a particular instance then it is also possible to use a type which is a child of it.

Whilst this feature may seem arbitrary it proves very useful in developing the current system. For example there needs to be a way of generating the mean of a set of data. All the users wishes to do is supply the object mean with the message containing the data and receive back the mean. Given there are a variety of different forms for the data including frequency and censored data all the users wishes for is the routine to calculate the mean. The mean is being used as a polymorphic concept.

### **5.3 Review of Definition**

A different prospective on computing is provided by object oriented programming. In this style of programming objects are defined in terms of the messages which they receive and which they respond with. The process of responding to a message is referred to as a method of object. The methods are the operations of the object and form part of the object. Only the methods of an object will change the state of the object. The object encapsulates the data and associated methods. From outside the object the procedures and data can be hidden and this provides protection for them. The objects are defined within classes, or hierarchies of objects.

### **5.4 Advantages and Disadvantages of Objected Oriented Programming**

Object oriented programming provides an elegant view of programming. The concept of an object seems very natural. It would therefore seem relatively easy to implement. However as was indicated by the key features there are implicit difficulties. How does one constitute a basic event within an object oriented system? There is the need to be able to differentiate between a message and an object which can prove difficult especially in the context of a statistical system.



Ensuring that an object satisfies the requirements of object oriented programming such as abstraction, encapsulation and independence can become fraught.

Balanced against these philosophical issues there are a range of pragmatic advantages for object oriented programming. Peterson (1991) claimed the following list of advantages:

‘they improve productivity,  
the use of objects as basic modules assists the designer to model complex real-world systems,  
the flexibility of object oriented code allows a rapid response to changes in user requirements,  
the reuse of standard components reduces both the development time for new applications and the volume of code generated,  
the increased maintainability of software makes it more reliable and reduces maintenance costs and  
object oriented programming encourages an incremental approach to software development.’

The main advantages of using object oriented programming, which are generally acknowledged to qualities desired in good programming, are modularity and reusability. Modularity is achieved when software elements, which correspond to syntactic units in the language, have explicit interfaces and provide a high degree of information hiding. A modular language enables the break down of complex problems into smaller problems which can then be solved separately. Modularity helps in the understanding of software and encourages the construction of new software elements by combining existing software elements in new ways. All these facilities enable the programmer to control the scope of both design and execution problems to a limited number of modules which makes the process of updating and maintaining software much easier. Given the role of objects within object oriented programming modularity is ensured.

## **5.5 Practical Issues**

Having considered the definition of object oriented programming there is also a need to explore the practical issues surround the paradigm. The structure of an object will depend clearly on the implementation. This will be effected by underlying structure of the language and will be described in terms of the implementation.

The term structure refers to the passive properties of an object, as distinguished from behaviour, the active properties. Every object has identity as part of its structure. The identity of an object distinguishes this object from all other objects, while allowing all references to this object to be recognised as equivalent. Objects are dynamically allocated whenever required. These properties are necessary for a language to be considered fully object oriented, Peterson (1991).

An object often refers to other objects. For example each element of an array is another object. Objects created refer to other objects through slots. Each slot contains one piece of data, more precisely, each slot refers to one object. A given object can be referred to by more than one slot, a slot does not contain an object, it simply contains a reference to an object, see Steele (1984).

Obviously central to the paradigm is the development of objects. Object creation requires a function that will create a new object provide its class and initialisation arguments that describes its desired properties. Initialisation arguments are keyword arguments. Each initialisation argument has a name ( a symbol ) and a value ( any object ). The initialisation arguments accepted by the function will vary, depending on the class. The value of an initialisation argument can be stored in a slot of the newly created object. The definition of a class specifies the names of such initialisation arguments and the slots they fill.



The programmer can describe the properties of a new object not only by simple filling of slots but also by writing initialisation methods. After creating a new object the function calls the initialisation generic function. Methods for the initialisation function receive the initialisation argument and perform any required initialisation. They can create subsidiary objects, enter the object into a registry of all instances of its class, extract some data from a database into slots, or implement additional arguments.

## **5.6 Comparison of Languages**

Object oriented systems originated from programming languages, such as SIMULA and SMALLTALK. In this section the variety of languages referred to as object oriented is explored by considering a number of implementations. The languages considered are CLOS, SMALLTALK and C++. Obviously the discussion will be restricted since it is not the objective of this section to describe in detail each of the languages. The objective is to give the diversity of the languages and their interpretation of the term object oriented. In any language the objects created will be determined by the nature of the programming language used.

### **5.6.1 CLOS**

CLOS (Common Lisp Object System) is an object-oriented programming language, embedded in COMMON LISP. It is therefore implemented on a wide variety of hardware platforms and operating systems. CLOS is based on objects with named slots, classes with multiple inheritance, and generic functions with method combination. CLOS provides encapsulation but not protection. CLOS emphasises power, efficiency, extensibility, and fitting smoothly into its host language, see Rumbaugh et al (1991). A user-defined class in an object-oriented language such as CLOS resembles a record type in a language such as PASCAL or ADA, Tello (1989). Since CLOS is embedded within LISP then LISP's built-in functions are available to CLOS.

### **5.6.2 SMALLTALK**

SMALLTALK is one of the oldest object-oriented language, originally developed at Xerox Palo Alto Research Centre on early 1970's as a research vehicle for implementing interactive systems and for studying the learning of programming, see Goldberg and Robson (1983). Smalltalk has been developed and implemented on a variety of hardware. It has been the inspiration for most LISP-based object-oriented programming languages and has been used for several real-world applications. However, it was not originally intended to be highly efficient or to support large programs.

In Smalltalk all objects have identity, a class and slots. There is no form of data other than objects. Storage is object-oriented and program flow of control is implemented with objects, giving an extra level of flexibility to the programmer and an extra measure of efficiency challenges to the implementor.

The principle dialect of Smalltalk uses single inheritance and contains no form of method combination other than shadowing. There has been few experiments with multiple inheritance extensions.

Smalltalk is not embedded in another language. Smalltalk has fewer built-in classes than CLOS; in Smalltalk almost all object types are implemented in the normal way available to users, not in a special underlying language. Smalltalk has no mechanism other than messages for invoking behaviour

Smalltalk has no significant protection or information hiding mechanisms. It has some introspection capabilities, but they were not as extensive or organised as CLOS meta objects. Smalltalk metaclasses serve an entirely different function from CLOS metaclasses, see Kim and Lochovsky (1989).

### 5.6.3 C++

C++ is an object oriented variant of the well known C language, see Stroustrup (1986). Like C which has been called a machine-independent assembly language, C++ places a very strong emphasis on efficiency, even when it impairs abstraction. Thus C++ avoids language features that require more than minimal support at run time.

C++ is not built on an object-oriented language like LISP, in which all data are in the form of objects, all objects have identity, and the type of any object can be determined at run time, independent of context. C++ provides both normal C data, which are not objects, and classes instances which are a special kind of record or (struct). In either case storage management is not automatic and must be done by the programmer.

C++ use single inheritance. The principle reason for this seems to be to ensure that the storage representation of a class instance is identical to the storage representation of an instance of any superclass, considering only the slots that exist in the superclass. Each subclass (called a derived class in C++) simply appends more slots to the end of the storage representation. Single inheritance allows positions of slots within instances to be compiled into methods as constants.

C++ uses messages (called member functions) to invoke behaviour. Most messages have only one method, so the method sending construct can be compiled as an ordinary function call with object as an extra argument. Virtual functions are messages that have more than one method and hence require a run time indirection through a class slot to determine which method to call. There is no form of method combination other than shadowing.

C++ offers overloading of operators such as +. Operator overloading is convenient alternate syntax for message sending; the first operand is the message receiver. In addition to messages, C++ also has ordinary functions. Ordinary functions can be overloaded, but unlike operator overloading this has no connection with messages.

Function overloading is a way to get behaviour that depends on object type, but the type must be known at compile time. Operator overloading can also be used alternate syntax for calling an ordinary function, see Kim and Lochovsky (1989).

Generic classes in C++ are a simple macro-expansion for defining several derived classes that are similar to each other but vary in some parameter. This does not involve any run time object typing. A generic class must always be instantiated to an actual class before a program can use it.

All these language features are designed to require little or no support at run time; except for virtual functions they are entirely implemented by the compiler's resolution of names to addresses.

To help enforce abstraction, C++ has facilities for name hiding. Slots, functions, and messages can be declared private, making their names available only in methods and friend functions. A friend function is an ordinary function that has been declared to have the right to access the private names of a class, see Wiener and Pinson (1988).

C++ contains no facilities for introspection. The compiler and the run-time environment are completely separate, and only the compiler contains the information required for introspection.

#### **5.6.4 Discussion**

Each of the above languages are object oriented, but in all cases some aspects of the language falls outside the strict definition of such. In CLOS and Smalltalk there is no protection for objects. C++ contains data structures which are not objects. In each case there are advantages to be gained from breaking from the strict definition. All these languages were considered for the development of the current system. However it was appreciated that they provided few utilities which would ease building of the functions required in the envisaged system. It would

therefore be necessary to devote time to building these utilities. The advantage of such would be utilities designed as required, but at the cost of time. It was primarily for this reason that it was decided to explore other languages which would offer at least some of the utilities required. However it was still desired that object oriented principles would be used. APL2 had many of the desired utilities required and so in the next section APL2 will be examined to see if it can conform to object oriented principles.

### **5.7 APL2 as an Object Oriented Language**

There are two categories of object-oriented programming languages, the languages which were designed to be object-oriented and already existing languages that could be extended to be object oriented. CLOS and Smalltalk are examples of the former and C++ is an example of the latter. Therefore it is possible to introduce object-oriented features to a language by establishing an approach to object definition using the structures available in that language. Obviously it will be harder for some languages than others to achieve the object oriented state.

For any languages to be accepted as conforming to object oriented principles it must be capable of supporting objects and the features described earlier in the chapter. It will be demonstrated that APL2 can deliver most of these features without too much distortion of the language. The first requirement is that APL2 should be able to provide an object type structure. These structures in APL2 are variables, functions and operators joined together in a workspace. 'Elements in a workspace are defined through assignment augmented by a collection of structure functions such as reshape and transpose, Frey (1992). Therefore the workspaces in APL2 are the natural medium to join together separate elements into groups where they can be referenced by name.

The creation of objects in APL2 can be achieved through creating local workspaces. In these local work spaces name references are employed locally with a set of external or public functions governing the communication with the parent



workspace or other subspaces by passing arguments and receiving results or vice versa. These can be interpreted as messages.

A simple assignment of one subspace to another in APL2 results in inheritance by reference for verbs. Firing a proverb in the NEW\_LIST makes the proverbs in the original list\_subspace to be called, but that execution will occur in the context of the NEWLIST\_subspace . To achieve inheritance by value we need to apply a fix adverb that will actually copy verb values from one subspace to another.

The difference between these two types of inheritance emphasises the desire and the ability in APL2 to control the type of inheritance, by reference or by value. This allows the introduction of subspace prototypes without introducing strict typing into the language. This is important since a key feature of APL is the lack of typing of variables. There is no need to declare a variable type it is defined by use. Multiple inheritance can be supported in APL2 by referring back to more than one prototype.

Subspaces add to APL2 certain useful facilities such as the ability to have verbs with static nouns (nouns with values that persist between calls), record structures (subspaces consisting only of nouns), program modules and libraries and simple objects such as our LIST\_subspace. All these facilities go a long way toward satisfying our original goals of modularity and reusability even if it does not achieve total object-oriented capability.

Returning to the features of object oriented programming abstraction is achieved by using subspaces since they are true abstracts of data. Users need only to concern themselves with the interface and not the implementation. Encapsulation is achieved by separating public and private names. The interface between a subspace and the outside world is clearly defined. A limited form of access is allowed through the new s\_conjunction. Inheritance for subspaces is handled as it is for any other object in APL2 that is by assignment augmented by appropriate structural operators. Finally polymorphism is a natural quality of APL2 because it

is not statistically typed, there is no need to employ special mechanisms which allow for type generalisation.

APL2 is powerful, but it is not particularly extensible and does not adequately support information hiding. These problems can make it unnecessarily difficult to design, implement and manage medium to large systems.

## **5.8 Discussion**

There are several reasons for adopting object-oriented programming in designing the system. It helps to eliminate redundant code, it provides a protection to objects from being invaded by code in other parts of the program. It is also time saving in being able to build programs out of standard working parts that communicate with one another, rather than having to start writing code from scratch. Finally, the most appealing feature is the ability to have as many instances of an object as desired copresent without any interference.

In this chapter having explored the features of Object-Oriented Programming we have examined the common OOP languages. In the last part we have considered the possibility of OOP in APL2. It was initially the intention to design the system as an OOP system. For this seemed to be the most appropriate discipline of design to work within. APL has been extended before to accommodate OOP design features, see Alfonseca (1990) and Frey (1992), demonstrating the power of APL in creating features such as Abstraction, Encapsulation, Inheritance and Polymorphism. These features were achieved with little restriction on APL. Having established that there would be granularity of design therefore it was envisaged that this design would conform to OOP principles.



## **CHAPTER SIX**

### **SYSTEM'S DESIGN**

## **6.1 INTRODUCTION**

In this chapter the design of the system is described starting with a review of the requirements of the system. These requirements will arise out of the desired features of the system. The aim as previously stated is to develop a system to aid the statistical analysis in the area of reliability. The system should offer the user statistical advice and assistance as well as the ability to implement that advice.

The first requirement for the system is a set of routines to carry out statistical analysis. These routines will have to be organised so that they can be used to achieve the aims of providing a systematic analysis of reliability data. This organisation should allow the routines to be driven by the user's objectives. The user should therefore be able to define their objectives within the system. Advice should be available for the user on the routines and on the analysis. Where possible the user should be given guidance on the analysis. This advice should be non-authoritarian.

The design should conform where possible to object oriented principles. This implies that there should be a modular structure containing objects which have the attributes described in chapter five. The language selected for implementing the system is APL2 and obviously will have its effect on the design of the system.

## **6.2 Desired Features of the System**

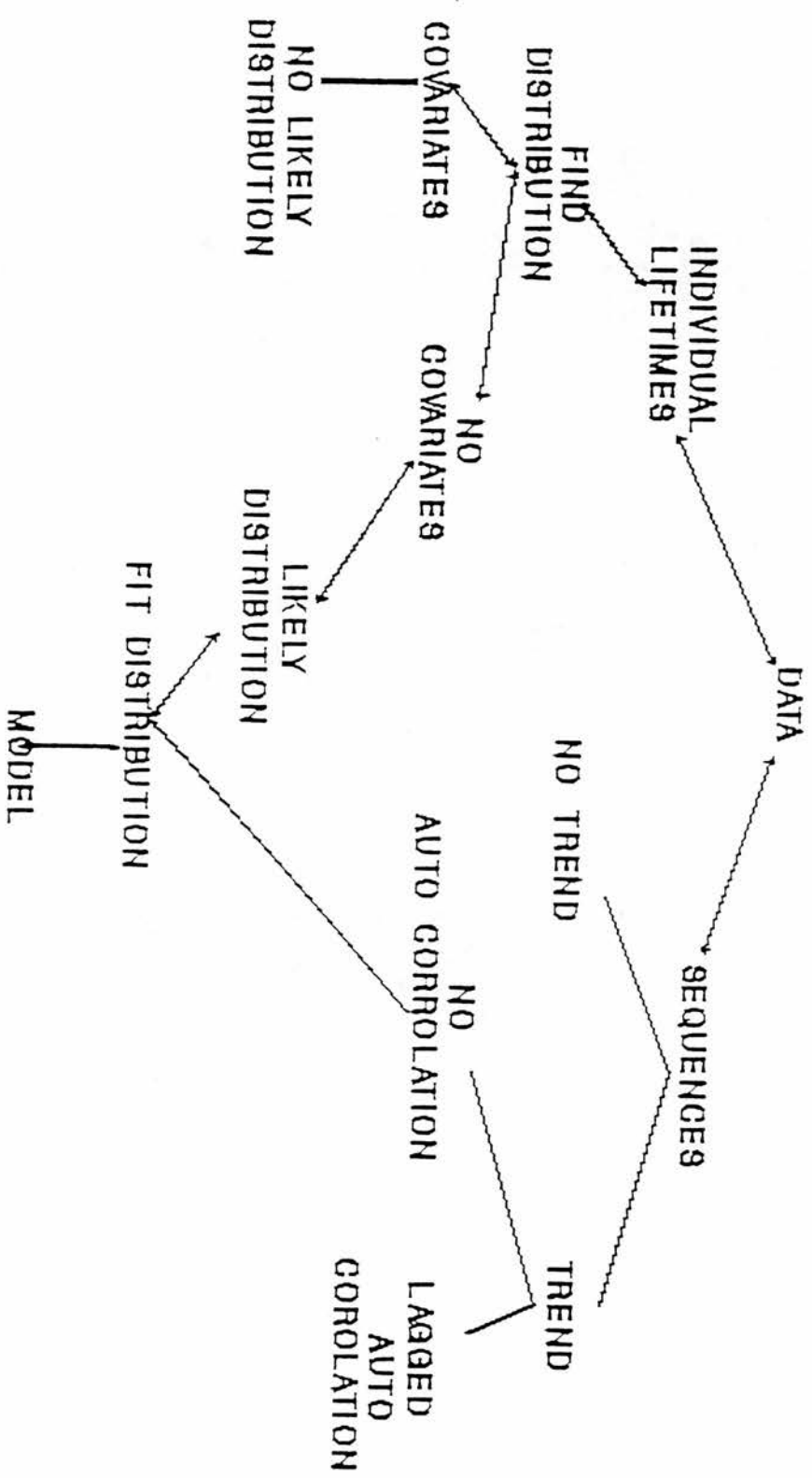
As a starting block the system needs a set of routines which will allow the analysis of reliability data as described in chapter two. Given that at the inception of the project there did not appear to be available an acceptable software package to deliver the routines, therefore these would have to be built. This, of course, allowed the flexibility in their design. However it was felt that a structure, or

organisation, for the routines was necessary. The structure should follow the shape of analysis. After considering the form of several likely analyses it was decided that a map could be constructed of the analyses. This subsequently became known as the analysis map and is illustrated in figure (6.1.)

The map should be treated as a road map allowing the user the freedom to chose where to start and where to finish. A particular analysis would follow a specific path through the map. There would be restrictions. It would not be possible for a user to analyse data without initially entering the data or its source. The map therefore would also suggest preferred routes for the analysis. Allowing the user to select though was seen as central to the idea of being non-authoritarian.

The paths on the map would form an analysis and so should meet some objective. Therefore it would be possible by specifying a user's objective in terms of a path to follow the analysis. This would provide the key feature of the system which is a problem led approach to analysis with the objective of the analysis being given a central role. The map therefore provides an ability to control the process within the system whilst retaining flexibility. If it is not controlled then the system may go through a series of undesired stages and processes to reach the required result. Satisfying the objective of an analysis will require the presence of a control mechanism which will assume a strategy or a set of strategies.

Considerable importance, though, is attached to the user's ability to control the system. The system should therefore not force the user to pass through a pre-determined set of menus of operation and processes in attempting to achieve their goal. The emphasis is on a knowledge enabling system which is characterised by its nonauthoritarian approach.



Figuer 6.1 - A simplified Analysis map for ARDA

This initial focus of the system stems from fitting a model to data. It is desirable to contemplate a more general system in which entry can be made from any of the elements Data, Model and Objective. It may be that the user wishes to know for a model what type of data is required or what 'Objectives' could be established. Similarly one could explore using either data or objective the other two elements. This flexibility of use if not contemplated in the original design would be difficult to incorporate at a later stage.

Moving between the elements Data, Model, and Objective requires a mechanism that operates like an expert system advisor, this is similar to the concept of statistical advisor in GLIMPSE. The expert system advisor is to assist the user not dominate the user's analysis. It is desired that this expert advisor should be non-authoritarian, that is when guiding the user through the system it should allow the user to make choices in analysis. The user may diverge from the advice given. Where possible if the user does diverge from the advice given then further assistance should still be given to the user. This can not be always guaranteed but it is desirable. For this reason it may be desirable to centre the advice on specific tasks. This is plausible in APL.

Therefore it is desired there should be at least two levels of assistance in the system. There is the high level guidance which is related to the objective and so to the analysis map. At a lower level there is a help to be offered on specific tasks. Besides these two level it seems appropriate to offer the user further information in the way of knowledge about tasks, models and other aspects of the system.

Having discussed the advantages of object oriented programming in Chapter five, it is desirable to design the system as an Object-Oriented system without restricting or inhibiting the system. It is an essential design requirement that the system retains a great deal of flexibility and lack of formalism to give the user the facility of

choosing the desired path of analysis and make full use of the help\guidance facilities available in the system.

In designing the system there was continual importance attached to giving the user the ability to control the analysis, ensuring the user has freedom where possible to achieve their desired goal without having to pass through a pre-defined set of menus of operations and processes. The system is expected not only to provide knowledge (help and assistance) but also the power to carry out techniques and methods. So it was decided to build the system as a knowledge enabling system and given these requirements there is need to select the appropriate design approach.

Whilst GLIMPSE may form a blue print in some senses for the current system it is also admitted that the concepts introduced by both Olford (1989) and Neave (1989) of a statistical environment is also an attractive model. The system therefore where possible should provide an environment for the user to work in rather than restrict the limit of the users capability. Again this is another argument for the use of APL2 since it provides the concept of the workspace.

### **6.3 The Design Structure**

The choice of using object-oriented programming resolves the philosophy of design. There, however, is a need to implement the philosophy within the system. Obviously central to the object oriented paradigm is the definition of objects within the system. This will be achieved by using a set of data objects with associated methods. In APL2 parlance the data objects will be nested arrays and the methods will be functions. The functions will operate on the nested arrays. These objects will as described by Frey (1991) be held within workspaces. The workspace can be thought of as the object. In doing this the system retains the

APL2 strengths of a range of mathematical utilities and the concept of a workspace environment which Neave suggests.

A global object, the system, with three modules Data, Model and Objective (children), and within these three modules we may have different structures (objects). For instance within Model there are two structures the stochastic models and the distributions. So the Model is an object which has children which are the stochastic models and distributions. Within the distributions there are families of related distributions.

The module Data has within it the Data vector which can be regarded as an object and it consists of other structures such as Data Status, Data Definition, Data Validity, Data Structure and the data itself. These structures are regarded as child objects of the object Data vector. The status vector is composed of the (objects) Data, Model and Objective which consist of other objects such as Data vector, Model vector and Objective vector. These vectors and all the procedures and functions constructed in them are considered as part of the object status vector.

Here a slight difficulty arises in the definition of the relationship of Data and Data vector with Model as objects and the definition of the message between them (the status vector ) as an object as well. As mentioned in chapter five treating every computer structure as an object is referred to as complete homogeneity where messages and properties (instance variables) are regarded as objects. This leads to circularity because if messages are viewed as objects then they should send messages to communicate with objects, and hence the ambiguous nature of messages arises. Similarly if attributes were objects then they should have their own attributes which are treated as objects as well. Therefore one has to stop at a certain level to break this circularity and the ambiguity which arises. It is then more accurate to describe the status vector as both an object and a message. This



enables us to clearly declare the movement of the status vector to a procedure and returning with a message. From an ontological view point such confusions is predicted.

Such formal specifications as in object oriented programming require decision on the nature of membership of objects in our system. It is a useful exercise since it gives a formalistic base in which to discuss objects, but this leads to contention in the definition. For instance as illustrated before, having defined the object Data which is associated with Data Input, the question is what is the relationship to the activity of model selection. If model selection is a part of Data then there are plausible problems of construction. It runs counter to the concept of granularity and affects the system structure which becomes a hierarchy. In such a system then it would be difficult resolving model aspects independent of data, this might restrict the system. Formalism will reduce the flexibility of the system. Whilst Formalism implies clarity it will reduce the degree of object orientation. A formal specification does not allow for the realities of programming and may inhibit design. Formalism is good where it provides a basis to ensure good programming. If good programming runs counter to formalism it is unfortunately the case that formalism should give way.

#### **6.4 ELEMENTS REQUIRED IN THE SYSTEM**

The system would consist of three main objects 'Objective', 'Model', 'Data' and two meta objects 'Control' and 'Knowledge'. These objects will conform to object-oriented programming principles where ever possible. Each object including the system itself would have a data structure, usually represented as a nested vector array.

In the case of the 'System' this would be the 'status vector' containing information about the current analysis. The object 'System' also obviously contains functions which act on the status vector. Some of these procedures will be contained within 'Control' and 'Knowledge'. The system would have children 'Objective', 'Model', and 'Data'. Each of these would have the possibility of further children. This section will describe the requirements for these objects, see figure (6.2).

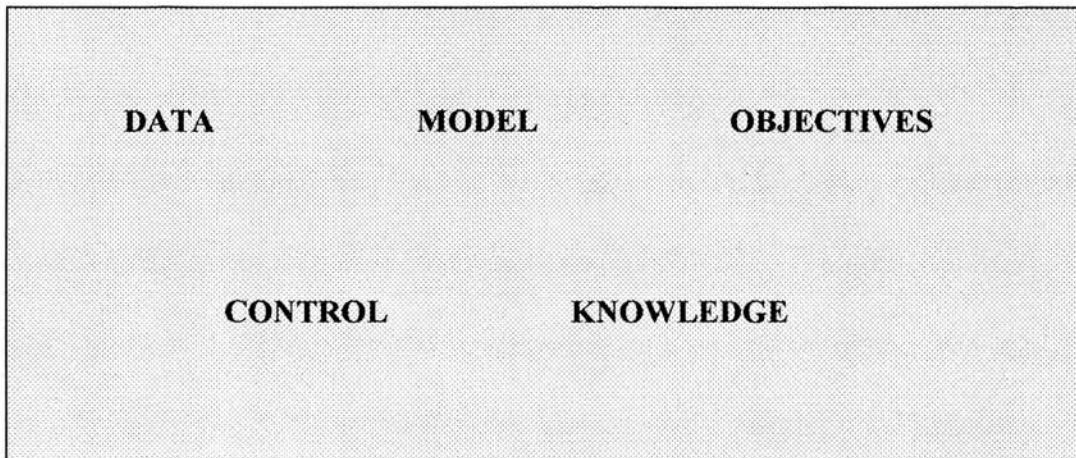


Figure 6.2 - The System.

#### **6.4.1 Data**

The main role of this object is primarily concerned with the entry of data but it also allows for data manipulation. This includes definition of the data entered and a description of its type and context. The process of data validation is also needed to determine the suitability of data for use in the system. The steps can be carried out adopting question / answer approach. Allowance should be made for the possibility of updating, checking and refining definitions. These definitions will influence the system's choice of path of analysis and certain paths will disappear,

see analysis map, see figure (6.1). For example if data consists of single lifetimes then it will not be normal to test for trend or dependency. Hence the analysis path is refined. It may be, though, that the user wishes for such data to be tested for trend. In such cases it is desirable for the system to adopt the approach of warning the analyst then allowing the user to do so.

#### **6.4.2 Models**

The role of this object is to define and fit the appropriate model to data. It should also enable the user to gain more information about the data models it incorporates. The first task which faces the user when analysing data is the need to establish whether the data model is known or not. If it is not known the user can use knowledge from the other system objects to decide the characteristics of the data model required. By interrogating the system's other modules such as Data and Objective, the user can go through a series of steps answering a set of questions thrown by the system to fit the right model to data. The process of moving from one module to another helps the user to define exactly what is required of the system and impose his/her will on the path of analysis. This process is influenced by the set of objectives achievable by the system. The user might be interested in whether the new component is better than the old one. Then the user is only interested in knowing if a component's lifetime is longer for the new component than the old one. This may require a parametric or non-parametric test. The user might be interested to know whether to replace or not to replace a component, i.e. whether components lifetime is getting relatively old. The question posed of the model is whether the distribution has an increasing failure rate or not.

#### **6.4.3 Objectives**

The system was designed with the user's desire in mind to determine the OBJECTIVE of analysis. In the objective dependent cases the user will interrogate the system to define a suitable path of analysis to his/her enquiry, see analysis map figure (6.1). This creates an interactive state between the user and the three main

system's modules DATA, OBJECTIVE and MODEL to reach the desired results. Hence it is possible to find for an objective either the conditions which require to be satisfied, the list of tests required to be carried to provide satisfaction or the strategy best suited to satisfy the objective.

The wide possibilities of objectives makes it difficult to cater for all possible objectives at this stage of system design. One interesting approach is the one adopted in INTERNIST, which is grouping objectives into families, and from these families select specific objectives. This also caters for the possibility of having several objectives arising from the same analysis. If the case was data dependent then the expert system will go through several stages of defining, validating and inputting data. Having acquired both objectives and data then the system will go through the stage of model selection.

Typical types of problems dealt with by the system would be:

- a) To assess the system's performance.
- b) To predict rates of events.
- c) to repair or replace.

Assessment of system's performance entails the assessment of some feature of the distribution and therefore assessing whether the performance is greater than some specified value or target, or the assessment of whether population A is better than population B. Alternatively it could simply mean the assessment of a feature such as distributional form, see figure (6.3). Adopting the INTERNIST approach the user has to go through a set of questions such as given in figure (6.3).



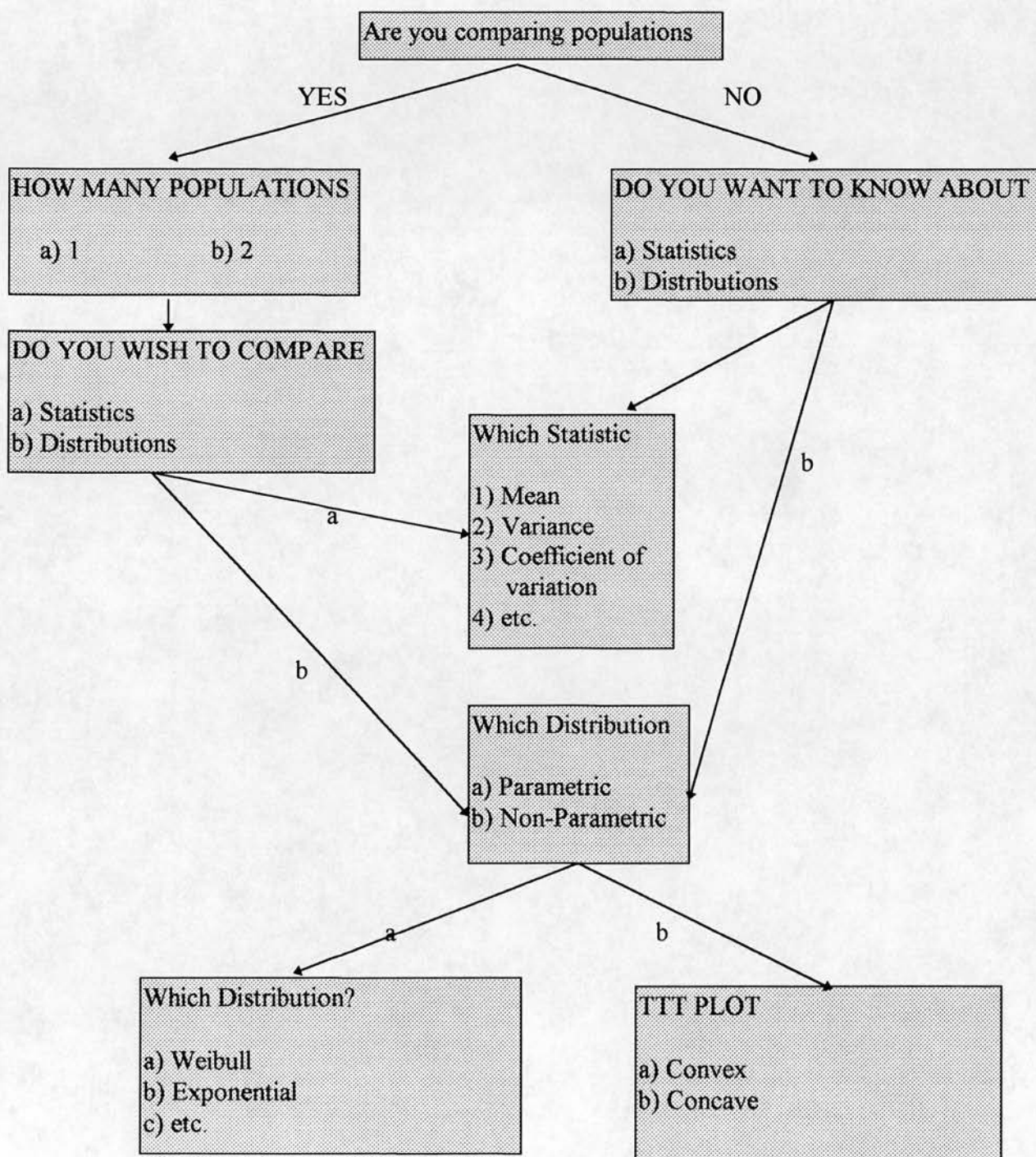


Figure 6.3 - Assessment of System's Performance.

Prediction of rate of occurrence is similarly very wide. Predicting the rate given some previous information which may or may not contain covariates and subpopulations. Rate may be affected by the stress the system is under or the environment the system is in. The rate may be all types of failure or particular form of failure.

For the purposes of the current research the decision whether to replace or repair an item is made solely on the basis of whether the component has a increasing or decreasing failure rate. Obviously it is point less to replace a component which is improving. Obviously a more sophisticated model could be developed using the costs of repair, replacement and failure.

The test whether a set of data indicates the distribution is increasing or decreasing can be achieved either for the parametric or non-parametric case. The non-parametric approach would be to consider TTT-plot, see Barlow and Campo (1975), and see whether it is convex or concave. A problem with censoring arises using this approach, see Lawless (1986) in the discussion of Ansell and Phillips (1989).

Obviously one might argue these limited number of options restricts the analysis. It was felt though that there was a need to demonstrate the approach could be employed and there is no reason why a greater range of options could not be used. Hence this is seen as a limitation of the system which can be easily overcome. It is also possible for the user to define their own objective and explore this using the map or alternatively the free standing functions.

#### **6.4.4 Control**

As was explained earlier part of the control mechanism is based on the analysis map presented in figure (6.1). Each of the processes, nodes on the map, indicated

on the analysis map is represented by a nested array that contains the name of the process, the tests that are required to fulfil that process and the names of the functions the user has to enter to invoke these tests, see figure (6.4). This is a type of hypertext, using the map and expanding it.

These nested arrays representing analysis map are all children objects of the nested vector Objective which will at end of session contain all the processes attempted by the user to achieve the objective of analysis. Given the user may have defined an objective then there will be a set of steps to explore to achieve the objective. This will provide guidance.

<b>HPP PROCESS</b>		
	NO TREND	TEST TREND
	NO DEPENDENCE	TEST INDEPENDENCE
	IDENTICAL	TEST EXPONENTIAL

Figure 6.4 - Nested array representing one of the processes on system's map.

The nodes of the path represent parts of the analysis. At a node a decision may be made and based on it the next node will be chosen. Hence in Figure [6.1] (the analysis map) each node will have one or more paths into it dependent on analysis and then a path or paths out. The choice of path will depend on the decision made at node, chosen either by outcome of analysis at node, pre-selected by objective or selected by the user. Again at each stage there is the desire to allow the user choice. A user may ignore the advice given, but will retain the information in the nested array for future use.

Control is the ability to direct the next action. Decisions on what the next action can be either proscriptive or open to the user. Given the desired aim of the system



to be non-authoritarian the system should allow the user to select from a range of possible alternatives. Obviously it could be helpful to the user if there was guidance about the next action. Proscriptive control will have to be adopted in certain areas and activities of the system where achieving the desired results is not possible without more information. For example it is not possible to estimate parameters without data. This will be achieved by exploring the relevant nested area to check the pre-conditions of the test.

#### **6.4.5 Knowledge**

The role of Knowledge in the system is both to provide a pool of general knowledge that might be useful to users with different levels of statistical expertise and also to provide guidance for specific tasks, for example data definition, validation, etc. Knowledge should consist of details of tests and when to apply them, also details on implementation of system functions and the steps involved to use certain function. It should also be designed to provide the user with information on types of data processed by the system and the tests and functions used in the Data module. It should also supply the user with advice on the types of models catered for by the system and the tests and functions involved in the Model module. Finally it contains a list of the objectives achievable by the system that the user can adopt when using the system.

The information will be incorporated in the system in different forms depending on the nature of process involved. In case of the module Model knowledge is presented in the form of a nested array that details the type of model, its parents, its children, its parameters, and the functions involved in that module. The same idea shapes knowledge on Data and Objectives.

The informal nature of APL2 should allow us to incorporate the required knowledge in the system concentrating on one function at a time. This makes the system's advice non-authoritarian giving the user the flexibility of exploring the knowledge available in the system freely without being restricted to having to follow the given advice, hence overcome the problem faced using GLIMPSE.

As for knowledge on functions in the system this consists of details on the name of the function, the nature of its input and output and the variables involved in its operation, the algorithms used, and the type of result expected. All this information should be available at a functional level being inserted at the beginning of each function and is accessed at the system level by a set of "HOW" functions.

As for knowledge on tests, a nested vector will contain all the information needed on the chosen tests with details that will assist in the interpretation of the results.

#### **6.4.6 Guidance**

Help facilities are one of the characteristics of expert systems and if designed with care they can improve the system's user friendliness. The system should provide Statistical advice when requested by the user to assist in choosing the appropriate path of analysis. It should also be able to explain its results, trap errors and suggest alternative patterns of analysis. Obviously the quality of this facility is directly proportional to the amount of effort invested in developing it. Since we are building a prototype the help facility has only been partially completed to illustrate what could be achieved.

As mentioned earlier there will be knowledge built within each function of the system. These lines of information are available to the user to access as a help facility to understand the system's functions. There should be a help function for all the major functions in the system. Each help function should be associated with a

major system function and is called when system function is used. These help functions will contain details on the statistical operation being performed by that particular system function. These functions allow the user the freedom of using the system to get statistical advice on Reliability of Data as well as understand the reasoning behind the system's decisions. At this stage only a limited attempt has been made to develop this help feature in the system.

In all types of help and advice the user should be given the freedom to reject the system advice and choose a different path of analysis and still be able to receive sufficient guidance and advice.

## **6.5 Discussion**

Much of the development work in information systems for management is aimed at improving the quality of human decision. In building this system the objective is to improve statistical analysis of Reliability data and aid the manager. The intention is to provide information to enable the manager to make a decision meeting a specified objective or set of objectives.

The system should ensure the quality of the information by data validation. In the current system design there is the computer modelling involved and the statistical analysis which hopefully will aid the manager to make better and more informed decisions. As mentioned earlier, the high level is the abstract problem solving stage, processing information to produce knowledge. Construction of rules, construction of guidance, thinking about how we build these objects, the expert systems. Then we come down to the low level where there is a need to process the user's information. This is the operational level involving processing the user's enquiry and manipulation of data. In a learning situation one hopes to be able to

marry both the operational knowledge gained to the development level, taking the examples or strategies developed by the user to update the system.

Given the defined area the volume of information covers the topics which are likely to arise in the sense that both models and technique are available. When building an expert system it is clear that the objective has to be known. There are two levels the meta level which is what type of information is being sought and then the operational level of the types of objectives. Although we have described the system as used for data analysis, it may be used to enhance the user's knowledge. The links between Objectives, Models and Data can be explored either to find what data is required to establish a given objective or which models are related to which objectives. Therefore it can provide a 'What-if' analysis which demonstrates APL's natural flexibility.

The current system provides an environment for statistical analysis in the area of reliability. It therefore conforms to Neave suggestions about statistical expert systems. The flexibility gained from the use of object oriented design and APL2 make this possible. Throughout the system there is the ability to build in information. Unlike system like GLIMPSE where the information is only supplied by the Statistical Advisor the current system provides assistance down to the functional level and a higher levels such as the nodes and in the whole analysis map. Beyond this there is the ability to use specifically stored information.

As indicated by Frey (1992) in Chapter five APL2 does not really satisfactorily provide the ability to hide aspects in the system except through the enclosed workspaces. This may be seen as an advantage since it is this feature which assist in the development of the statistical environment. In the subsequent chapter the implementation of the system will be considered.

## **CHAPTER SEVEN**

### **THE SYSTEM**

## **7.1 Introduction**

The aim of this chapter is to describe the implementation of the system, hereafter referred to as ARDA (Analysis of Reliability Data in APL2). Having contemplated the different design approaches possible in chapter four and decided to adopt object-oriented programming techniques in building the system, a description is given in this chapter of the software. Details of the different modules (objects) of the system, their structure and main functions are included. To illustrate the system a typical session is included in this chapter.

## **7.2 ARDA**

### **7.2.1 Background**

ARDA's goal was to provide assistance in the analysis of Reliability data. This is interpreted to mean provide both routines for analysis and guidance on the analysis. As we have previously considered the manager / decision maker is not concerned with the statistical analysis but with the conclusion reached. A manager may, for example, wish to know whether a replacement strategy is worth employing without having to determine the type of distribution the data has. The system will therefore focus on the objectives of the analyst.

In designing ARDA effort was made to separate functions into two levels. The user employs the higher level functions, the strategic level functions, in which are incorporated lower level operational functions where the statistical operations and techniques are implemented. The system has been designed assuming the user is knowledgeable about Statistics. It is hoped subsequently to develop a system for the statistically naive.

### **7.2.2 Description of ARDA**

In chapter five the advantages of Object Oriented Programming and the suitability of APL2 as a programming language for Object-Oriented Programming, and in chapter six the ambiguities thrown up by such design structures were considered. ARDA therefore cannot be wholly said to be Object-Oriented, though Object Oriented principles have been implemented where it was useful and possible .

ARDA is an object and is composed of objects such as Data, Model and Objectives. This structure ensures a granularity of design. Each object will have a data structure which will be a nested array. In ARDA's case it would be the 'status vector' holding information about the current analysis. Each object has procedures which operate on it. In the case of the 'system' object, these procedures act on the status vector incorporating knowledge and control in the system.

The granularity is reinforced in the implementation by dividing ARDA into several work spaces. For example there are the workspaces `Rdadist.apl`, `Rdamodel.apl` and `Rdaobjective.apl`. These respectively describe the distributions, the Stochastic models, and the Objectives, see (Appendix 1) for more detail. This Modularity of structure improves the system's user-friendliness and provides it with a clearer structure. Flexibility arises out of the ability to incorporate more than one work space in the same analysis session.

Finally the modularity in ARDA's structure eases the process of updating and adding new functions to the system. Different functions of ARDA are reused in more than one workspace for example Mean, Stdev, Stats, the status vector which acts as a message between the different modules of the system. It is simple to overwrite a function in a work space and then copy to other work spaces. This enables the user to load the required functions for a specific analysis process in one work space making that workspace independent in its needs from other workspaces. It is also possible to load more than one work space together to form a single larger work space capable of achieving more than one type of analysis.



This facility enhances the system's flexibility and focuses the software on specific tasks without the presence of unnecessary functions as the case would be in one large work space. This creates granularity in software design and conforms to Object-Oriented principles.

### **7.3 Statistical Elements of ARDA**

The structure of the system consists of three main components,

#### **7.3.1 Statistical Advisor**

GLIMPSE introduced the idea of a statistical advisor which provide information about the techniques employed in GLIMPSE and on the next step to take in the analysis. In the current system the statistical advisor is an abstract concept, it exists in the shape of statistical help and knowledge incorporated in the system at different levels. For example knowledge is incorporated in each function describing the nature of the Left Argument and Right Argument of the function, the role of that function, the nature of the algorithm employed and the type of result expected. There is also information about the functions of the system, such as a set of HOW functions that will explain with examples the way each function should be used. Help and advice is made available at the nodes in the analysis tree. The form of the knowledge about models, stochastic processes and distributions is in nested vectors.

#### **7.3.2 A set of Statistical Models.**

Central to the development of ARDA is a set of statistical functions to allow reliability data analysis. These range from basic statistical functions to calculate such measures as means, standard deviation etc..., and more sophisticated functions to fit distributions and estimate statistics of such. Obviously given that data arises from a range of stochastic processes, there is a need also to be able to model and describe these.

In order to fit a distribution first there is a need to understand, where appropriate, the stochastic process and then fit a distribution to the lifetimes arising out of the process. In many cases the distribution to be fitted will be known. Following McDonald and Richards (1987) a hierarchical set of distributions is contained within ARDA. These are Weibull, Gamma, Normal, Lognormal, Chi, Exponential,  $t$ , and Poisson distribution. Obviously there will be contexts when it will not be possible or desirable to fit a known distribution but to use non-parametric methods or semi parametric methods. However it might be noted that in Reliability estimation is usually carried out for field data in the presence of high levels of censoring. Therefore specific function were developed in ARDA to deal with right censoring. The estimators provided are in most cases maximum likelihood estimators. A distributional free estimator of the underlying distribution function is provided, the Kaplan Meier estimator.

Over the recent past, stemming from Cox's work, see Cox (1975), there has been considerable interest in analysing lifetime data in the presence of covariates. These covariates may be either design features or represent the context of the component or system. The area has become known as survival analysis. In the following section more detail is given.

Whilst in most cases it is worth exploring possible distributions to fit as a first stage there may be more structures to data which needs accounting for. The nature of the data will dictate what approaches are possible.

### **7.3.3 Survival Analysis**

The aim of survival analysis usually is to relate the survival chance of a component to some feature of its design or experience, the covariates. There are other reasons advanced, for example Ansell (1987) suggested the reason may be,

- (a) to find significant factors (or variables ) which affect lifetimes;

- (b) to remove nuisance variables which distort analysis;
- (c) to increase comprehension of the failure model; and
- (d) to produce a better prediction of the failure rate.

The interest in relating lifetimes to covariates was rekindled by Cox (1975) who suggested a semi parametric method of estimation referred to as Proportional Hazard or Cox-Regression. There are alternative models such as accelerated failure time models, see Nelson (1993), and Weibull Regression model, see Smith (1991). The Proportional Hazard model has been used extensively in Reliability, see Ansell and Ansell (1987), Newby (1994), etc. Details of the approach are given in Ansell and Phillips (1994).

Regression models are used to measure the variation in one variable in relation to other variables or factors. The type of distribution of lifetime will affect the algorithms used for estimation. Given the variety of possible models it is important to choose a suitable one to be able to detect and assess the effect of variables which influence lifetime strongly. Other factors with a weaker effect can be ignored assuming the level of censored observation is negligible, see Ansell and Phillips (1994).

#### **7.3.3.1 Proportional Hazards Model**

Although Cox's proportional hazards model is considered to be robust, care must be taken when applying it to data. A number of diagnostic graphical techniques based on the residuals are available. The analysis is based on Cox's suggestion of partial likelihood in which the risk set at time of failure is considered. As a regression model it suffers from the usual drawbacks. Some of these can be tackled by use of plotting techniques and diagnostic tests. A recent summary was produced by the Treasury (1995) reflecting several statisticians' views about the recommended techniques to fit Proportional Hazard model and concluding that there are two techniques that can be recommended. The first is modeling variation

with time and assessing improvement in goodness-of-fit. In this technique it is recommended that time is divided into distinct epochs and a separate proportional hazard model is fitted to each epoch. The covariates are multiplied by a suitable function of time. The second technique is Diagnostic Plots. The summary restrict them to two types the first is stratification by covariate and plotting either the logged hazard or the logged cumulative hazard. The second is Residuals Plotting which is plotting Martigale or Schoenfeld residuals.

## **7.4 The Structure of ARDA**

The general structure of the system is object based with nested vector arrays acting as the data structure. The system consists of five elements, these are,

- 1- Data
- 2- Model
- 3- Objective
- 4- Control
- 5- Knowledge

the first three have associated with them vector nested arrays. 'Knowledge' and 'Control' will consist of procedures based on the status vector.

### **7.4.1 Data**

This object is primarily concerned with the entry of data but it also allows for data manipulation. The procedures operate on the vector nested array 'Data'. This vector consists of the following elements, see figure(7.1):

<b>STATU S</b>	<b>DEFINITION</b>	<b>VALIDITY</b>	<b>STRUCTUR E</b>	<b>X</b>
--------------------	-------------------	-----------------	-----------------------	----------

Figure (7.1) - Data vector nested array

The elements of the vector are defined as follows:

'**Status**' indicates whether data has been defined or not.

'**Definition**' defines the form of the data, single lifetimes or aggregate data, single component or sequence of components, covariates present or absent, whether the data is censored or not.

'**Validity**' specifies what the data was collected on, the component parts of the data, serial numbers, where collected, the physical conditions and any peculiarities as well as the acceptable ranges of values.

'**Structure**' describes the form of the data, size of array for the variables and form of the 'X' nested array. 'X' contains the values of the data.

Associated with the separate elements (objects) will be a set of procedures (functions). Some of these functions will be composed of simpler functions. For 'Status' there exist the function **INITIALD** which will set up the data array and set 'Status' to 'Not Defined'. **DTWOALL** is associated with '**Definition**'. It is composed of 5 functions each associated with an element within array 'Definition'. Similar functions exist for the other arrays: '**Validity**', '**Structure**' and 'X'.

#### **7.4.1.1 Data Definition**

This process involves defining the type of data being analysed. It includes several functions that will interrogate the user on details of data. This information will be held in a defined position in the Data vector nested array to be used in analysis. The system is capable of defining single values, single or double dimension arrays. The function dealing with data definition interrogate the user on information on data.

It is also possible to display parts of the data having completed entering it using the **DISPLAY** function. This function provides the user with the facility to display a single number, a set of single values or an array.

#### **7.4.1.2 Data Validation**

Data validation is achieved using the function **DATMOD** which acts on the data and specifies what the data was collected on, the component parts of the data, serial numbers, where allocated, the physical conditions and any peculiarities as well as the acceptable ranges of values.

#### **7.4.1.3 Data Input**

In the current system there is a set of APL2 functions which enable the user to input data with ease due to the flexible nature of APL2 and its ability to handle different types of data effectively.

The size of the sample data is an important factor in choosing the appropriate model. Although it is possible to eliminate some models from consideration, a small sample size may cover several different distributions, but there could be a lot of difference between tail probabilities from these distributions. Larger samples may require the characteristics of a specified model. In ARDA it is possible to adopt testing techniques which will cater for a variable size of data and produce reasonably accurate results. This is made possible due to the ease in programming provided by APL2. These testing techniques can cover all methods of Goodness-Of-Fit testing.

#### **7.4.2 Models**

The vector nested array 'Model' contains details of the current model. Its format is as follows, see figure (7.2).

The elements of the vector are defined as :

**'Status'**            Indicates whether the model has been defined or not.

**'Stochastic'**    Describes the form of stochastic processes concerned.

**'Distribution'** Names the underlying distribution.

<b>STATUS</b>	<b>STOCHASTI C</b>	<b>DISTRIBUTION</b>
---------------	------------------------	---------------------

Fig. (7.2) - Nested Model Vector

Other than that there are a set of vectors referring to the distributions and the stochastic processes, see figure (7.3).

<b>NAM E</b>	<b>PARAMETER S</b>	<b>PARENTS</b>	<b>CHILDRE N</b>	<b>ESTIMATIO N</b>
------------------	------------------------	----------------	----------------------	------------------------

Fig (7.3) - Distributions vector

The elements of the vector are defined as follows,

- 'Name'** is the name of the distribution.
- 'Parameters'** provides the parameters of the distribution as described in Macdonald and Richards (1987).
- 'Parents'** indicates the distribution(s) immediately above the distribution in Macdonald and Richard's tree of distributions.
- 'Children'** indicates the distribution(s) which are immediate descendants of the distribution.
- 'Estimation'** names the function(s) which will estimate the parameters of the distribution.
- 'PDF'** names the function(s) which will calculate the Probability Density Function.

Functions were developed to produce the tree of distributions given by Macdonald and Richards (1987) from either 'Parents' or 'Children'.

Similar vector exists for stochastic processes, see figure (7.4).



<b>MODEL A</b>	<b>DESCRIPTION</b>	<b>RELATIONSHI P</b>	<b>ESTIMATIO N</b>
----------------	--------------------	--------------------------	------------------------

Fig (7.4 ) - Stochastic Processes vector.

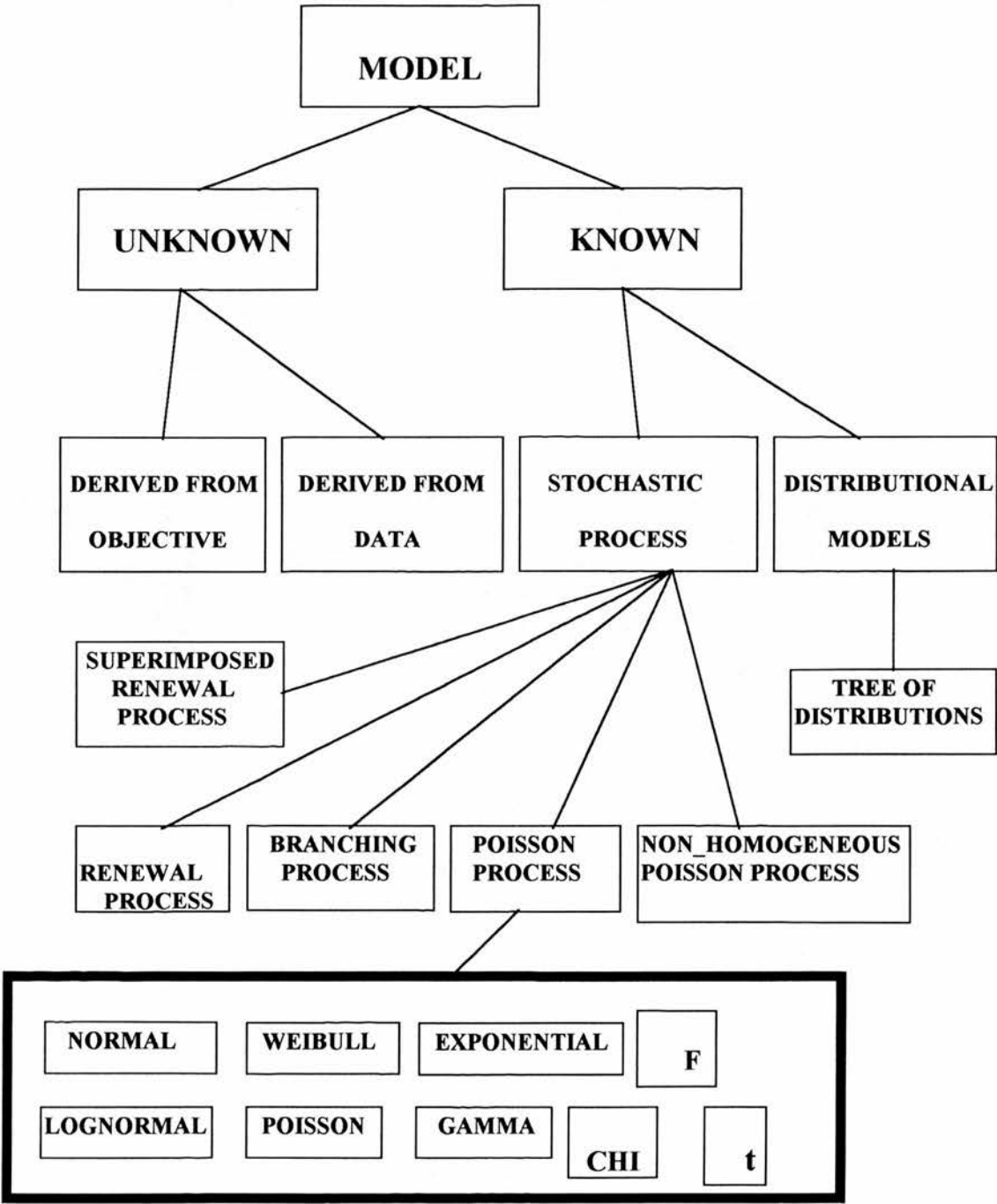


Fig (7.5) - ARDA’s Model Selection Process

#### **7.4.2.1 Data Model Selection**

It was possible to achieve model selection relatively easily by following the decision tree in Macdonald and Richards (1987). Several functions in ARDA are used to choose between the different distributions presented in the diagram. As mentioned earlier the system can deal with a set of distributions and a set of general stochastic models, (see figure 7.5).

#### **7.4.2.2 User's Desired Model Selection**

ARDA provides the user with the facility to select the appropriate model for the type of analysis the user desires. The user is therefore in control of the analysis. For it is neither practical nor appropriate to force the user through a series of undesired stages and process to reach the desired outcome. The system will ask the user to define his/her desires to establish a set of objectives.

The system will then proceed to explore the data or models with the objective guiding the analysis through the analysis map, (see system map 2 figure 7.6). The user will be offered menus at the nodes of the map which will allow intervention. The menus will supply the user with information and in some cases advice or guidance. This is the practical aspects of the abstract concept of Statistical Advisor.

The Statistical advisor will connect the desires, objectives and type of data together to decide on the right test and explain why if requested. The user then can inquire about the type of models which could be considered. A decision is made depending on the type of data on the model or models suitable which will fit data. These models will then be tested to see which will fit and which objectives are satisfied at that stage.





### **7.4.3 Objectives**

The main task of the 'Objective' vector is to define and clarify the objective, and hence specify the strategy. As with the distributions in the object 'Model', for each of the objectives there will be a vector. This will contain the set of conditions for the objective to be satisfied. These conditions will be related to tests on the data. The tests for a particular objective will form a path in the analysis, (see system map 2, figure 7.7). Each objective is represented in ARDA by a nested vector of three objects, the objective itself, the process required to achieve it and the name of the function needed to carry out the process, (see figure 7.8).

Following an INTERNIST approach in dealing with objectives, the current version of ARDA caters for the following groups of objectives:

- 1- To repair or replace
- 2- To calculate data's statistics
- 3- To test for trend
- 4- To fit a distribution to data

#### **7.4.3.1 To Repair or Replace.**

As mentioned in chapter four the user needs to decide whether to repair a simple system or replace it. The data will consist of a set of events and an assumption of instantaneous repairs is made. The user needs to establish whether a component has an increasing or decreasing failure rate. The decision to repair or replace is made if failure rate is increasing.

OBJECTIV E	PROCESS	FUNCTION NAME
WHETHER TO REPAIR OR REPLACE	FIT DISTRIBUTION TO DATA	FIT-DIST-ALL
	CALCULATE MAX-LIKELIHOOD OF AVAILABLE DISTRIBUTIONS	SDEV
	SELECT HIGHEST LIKELIHOOD	INDF

OR NOT		
--------	--	--

Figure (7.8) - A vector nested array representing the objective repair or replace.

The first step is to establish what form of distribution data has and to achieve that the function FIT\_DIST\_ALL is invoked and should return with the appropriate distribution for data. Then the maximum likelihood of available distributions is calculated using the approximation of the Standard Deviation. Then the highest likelihood is chosen using the function INDF which will decide whether failure rate is increasing or decreasing. If failure rate is increasing then a decision to repair or replace is made, see figure (7.8).

**7.4.3.2 To Calculate Data’s Statistics**

This objective is useful to assess the performance of a single population against a fixed rate or to compare several populations. The function STATS is invoked and the user is then provided with the minimum value, 25 pctile value, median value and 75 pctile value. Also The function MEAN to calculate the mean, the function SDEV to calculate the standard deviation, VAR to calculate the variance and CV to calculate the coefficient of variance, see figure (7.9).

OBJECTIVE	PROCESS	FUNCTION
CALCULATES DATA’S STATISTICS	CALCULATE VALUES	STATS MEAN SDEV VAR CV

Figure (7.9) - A vector nested array that will test for Data’s statistics.

**7.4.3.3 To Test for Trend**

Data is tested for trend using the function TRENDHM .

**7.4.3.4 To Fit a Distribution to Data**

This objective is achieved by calling the function FIT\_DIST\_ALL which will find the appropriate distribution to data.

**7.4.4 Control**

Control is the ability to select the next action. Decision on what next action can be proscriptive, or open to user.

**A- PROSCRIPTIVE**

This is taken to mean the system indicates the next stage. ARDA though is built to ensure that the user decides whether to accepts system’s suggestion for the next stage or choose a different process. For example if the user has tested data for trend and the answer was negative, the system will suggest to test for independence.

**B- OPEN TO USER**

This type of Control mechanism allows the user to select the next stage. For example on entering ARDA the user is asked about the type of investigation to be carried out:

Do you want to investigate

Data            Model            Objectives

At this stage of system development we will allow the machine to guide the user through a session which can consist of several enquiries on different sections or modules of ARDA and where several activities can take place. Therefore at the end of every particular enquiry, section, or activity there should be a set of menus that offer the user a limited choice. For example, at the end of an investigation which has involved the section Data of ARDA, on leaving the section the user is given the choice to redirect the investigation to another section of ARDA such as



Model , Objective, return back to the main system menu, to specify another direction or to quit ARDA. The message returned by ARDA is,

DATA - EXIT - To which part of the system do you wish to go,

Model

Objective

System

Other

Quit

Another example of using menus in ARDA as a control facility is the choice given to the user when selecting a certain activity in ARDA. For example in Model the user is given the choice of several steps involved in model selection, the option to describe other tasks for the system to carry out, the option to return to the main

system's menu or to Quit in the following format:

Activity Model - Definition of distribution,

Do you wish to - Define Stochastic Processes

- Investigate properties

- Estimate Parameters

- Main Menu

- Other

- Quit

In both cases there is an additional option which is 'Other'. This allows the user to move to a different section.

Proscriptive control will have to be adopted in certain areas. If an activity cannot be achieved for lack of information, then the system will interrupt the analysis session. For example, we cannot estimate parameters of distributions without having the data. As mentioned earlier, if we have an objective the analysis will

follow a path through the analysis map. A series of steps will be set up. Control is therefore through the paths. However, it should be remembered that the results obtained at each stage will also have an effect on the analysis. In some cases it will reroute the analysis; in other cases it will bring the analysis to an end.

Each test on the path will have a low level function and a high level function. The low level function will simply calculate the test statistic. The high level function will act as the driver for the test it will follow the following form,

- 1- Check\_Test
- 2- Test\_Statistic
- 3- Compare\_Test
- 4- Report\_Test

Check\_Test will check that the pre-conditions for the test exist. For example, it will test whether the data exists and if it is of the right form. If the pre-conditions are not satisfied then a report will be made which will include details from the low level function of the test statistic to be calculated. The Test\_Statistic function will be polymorphic dependent on the form of data. Compare\_Test will compare the test statistic with the appropriate distributional table.

Based on the outcome of compare\_test, report\_test will produce an interpretation of the results with advice and guidance on the next stage of the analysis. This will, of course, take into account the current objective. It is assumed that the advice is non-authoritarian and it is left to the user to decide whether to accept the advice given or run their own analysis. For example if we wanted to test Data for trend

**Command** - T\_Trendlr X

**Response** - If there is trend response will be -

" There is trend - Fit Non Homogenous Poisson Processor Model, or Other Stationary Model such as Crow's Model."

If there is no trend the response will be -

" No Trend - Test for Independence or Review objective of analysis.

T\_Trendlr is the function which will test for trend and decide whether there is a trend in data or not. It is the driver function of four other functions, the first is CH\_Lptrd this function checks data for trend, to do that it calls another function S\_Lptrd which calculates the laplace trend of a set of distributions that the system caters for. Then a comparison of likelihood using the Probability Density Function is done by the third function CO\_Lptrd. The PDF is calculated for the above set of distributions and the one with the maximum PDF is chosen being the most likely. Finally the function R\_Lptrd reports the results in the final report of analysis. The function T\_Trendlp is the driver function which is called by the user while the other four functions are operational functions hidden from the user. If there is trend in data the system will suggest to fit an Non-Homogenous Poisson Process model or another stationary model. Having come out of T\_Trendlp the system will provide a set of options.

#### **7.4.5 Knowledge**

The role of the object Knowledge in the system is to provide a pool of general expertise that might be useful to users with different levels of statistical expertise, and also to provide guidance with specific tasks such as Input, Data Definition, etc.. Knowledge is provided in ARDA on three separate levels. The first is information about interpretation of analysis results. This information is included at functional level where the user is guided at each node to the possibilities caused to by the results reached at that particular node. These interpretations are explained in the system's function listing (Appendix 5).

The second level of knowledge provided in ARDA is information about ARDA's different modules, Data, Model, and Objective. This knowledge is made available to the user in the form system description in the user guide and numerical examples

(Appendices 1 & 3), also in the form of diagrams and tables and system maps detailing each possible path of analysis.

The third level of knowledge is stored in nested arrays related to each system module such as the nested array Model which include information on type of model, details on that particular model, relationships to other models and estimated parameters of that model. The nested array Data also contains knowledge on data, such as whether it is defined or not, and the type of data is it. It also includes data itself.

There are also the distributions vectors, based on the tree of distributions given by Macdonald and Richards (1987), these vectors hold knowledge on the different models of distributions, (see section 7.4.2 earlier). The elements of a distribution vector are the name of the distribution, its parameters, its parents indicating the distribution(s) immediately above the distribution in Macdonald and Richards tree of distributions. The vector also holds knowledge on the distribution children, that is the distributions immediately descending from the distribution, the name of the function in ARDA which will estimate the parameters of the distribution and finally the name of the function in ARDA which will calculate the probability density function. This information is easily accessed by the user.

There is room to improve the availability of knowledge to user by creating nested vector of knowledge on tests interpretation and improve ARDA's guidance for the statistically naive users.

## **7.5 A Typical Session with ARDA**

Due to the Object-Oriented nature of ARDA it is possible to start a statistical analysis session with ARDA starting at any of its three main modules (objects) Data, Model and Objective.

As mentioned earlier ARDA's functions are divided in seven separate workspaces. These are **Rdastart**, **Rdadata**, **Rdadis**, **Rdamodel**, **Rdaobjec**, **Rdaknowledge** and **Rdaguide**. On starting a session with ARDA the user is expected to load the workspace **RDASTART** by enter the,

**command**        )**LOAD RDASTART**

**Response** A menu giving the user four options as a starting point to the analysis

session. The menu looks like the following,

1 - DATA ANALYSIS	NOT TESTED
2- OBJECTIVE ANALYSIS	NOT TESTED
3- MODELS ANALYSIS	NOT TESTED
4- STATISTICAL KNOWLEDGE	NOT TESTED
5- QUIT	

The " NOT TESTED " message which appears opposite each menu item indicates that it is a new session and that none of the menu items have been selected before. The user will need to enter the number of his/her choice and depending on that choice ARDA will instruct the user to copy the appropriate workspace in. For example if the user entered number one to start data analysis the response will be

      )**Copy Rdadata**

When the above command is entered all objects in the workspace **RDADATA** are copied into the current workspace.

Any analysis in statistics normally starts with an understanding of the data. Besides comprehending the context, this implies an initial data analysis stage requiring descriptive and summary statistics.

If the user wanted to check whether data is available or not, this is easily determined by checking the status vector, **D**

**Command**    Data1

If data is not defined then

**Response**   " Data is not defined "

If data is defined then

**Response**   " Data is defined "

The user is instructed to start data analysis by entering the **command**    Data

At the start of the session the user is interrogated by ARDA to define data.

**Q** " Is Data new data (New) or old data (Old)?"

If data was old data then the user is asked whether data needs to be updated or not. The user has the facility to add, change and erase items of data by indicating its position and value. If data is not known, it is new data, then it should be explored and the user will be interrogated by the system on details of the data to establish the type of data, its form, its context, whether it consists of one, two or more samples. The questions answered by the user will also determine whether data is multivariate or univariate. The user will then enter data or define a name that represents data in memory.

ARDA starts data definition by asking

**Q** "Does Data consist of single (Single) or Multiple (Multiple) lifetimes".

The user of course will enter one of the answers in brackets.

**Q** "Are there any Covariates ? Y/N"

if the answer is yes

"Enter number of covariates"

**Q**"Is Data dependent or independent? Dependent/Independent?"

**Q** "Is Data Lifetime data(Life) or Aggregate Data(Agg) , I Don't know (Not) ?"

if it is lifetime data then ARDA will respond with the following

question



**Q** "Is Data a set of lifetimes (Life ) or is Data a set of failure times (Failure) ?"

if it is aggregate data ARDA's response will be

**Q** "Is Data a number of given intervals (INT) or

Is Data a total at given times (Tot)

If you do not know enter (Not) ?"

**Q** "Is Data a single component data (Component) or

Is Data a sequence data (Sequence) ? .

if the answer was single component data then

**Q** "Is Data a covariate data (Covariate ) or

Is Data a no covariate data (No) ?"

if the answer was sequence data then

**Q** " Is Data a single sequence (Single) OR

Is Data a multiple sequence (Multiple) ?"

**Q** "Is Data censored data (Censored) or

Is Data not censored data (Not) ?"

**Q** " What was the data collected on ?"

**Response** The user can list the names of the objects which data describes such as aircraft, or factory machines, etc..

**Q** " What are the components of Data ?"

**Response** The user can enter the details of the different parts of the main object which data was collected on, for example in the aircraft example the response for this question would be to list all the names of the components of an aircraft such as jet engine, tyres, etc..

**Q** " What is the serial number ?"

**Response** Alphanumeric record describing the serial number of main object of data.

**Q** " Where was data Collected ?"

**Response** The location of site such as Edinburgh Airport in the aircraft example or factory or field where data was collected.

**Q** "Does Data has any physical context ?"

**Q** "Are there any peculiarities in Data?"

The answers of these questions will form part of the data report.

However, let us assume initially that some specific data is available and that it consists of a failure time with a set of covariates. The failure time may be censored time. For example aircraft engines, time to failure of engine has been recorded with analysis of oil for metals and details of routes. The first column of the data will always be failure time. Second column should indicate whether data is censored ,0 - not censored and 1 - censored. The remaining data would be covariates with no particular structure. Where there is a defined maximum time this should be recorded. All measured variables should have a range of acceptable values which will be checked during the data validation process. Design/context variables should always be discrete (integer) with a specified range of values.

After this initial stage of data definition the user is given the choice of starting another action by being presented with the main menu. The user will notice that the "Not Tested " message declared opposite to Main Menu option (1) has been replaced by " Tested " message.

One of the unique features of ARDA is that it gives the user the facility to declare his/her objective of using ARDA. To do that we copy the workspace dealing with objectives into the current workspace by entering

**Command** )Copy Rdaobjec

**Response** A list of objectives achievable by ARDA

As mentioned earlier, ARDA currently includes a limited set of objectives. These are generally management question such as "should we replace the current component with a new one ? " or "Is a component better than another?".

Having decided on the Objective of analysis, the user is returned to the Main Menu.

To be able to realise whether it is possible to achieve these objectives or not we need to fit a model of distribution to data. Assuming the data is a set of lifetimes, then there is a decision to be made about which distributions to fit. Whilst in most cases it is worth exploring possible distributions to fit as a first stage, there may be more structure to the data which needs accounting for. The nature of the data will dictate what approaches are possible.

The user will select option three of the main menu which is "Model Analysis" and ARDA will respond by asking the user to enter the command

**Command** )Copy Rdamodel

This will copy all objects from the workspace Rdamodel to the current workspace.

**Response** Then the user is asked to enter

**Command** Models

**Response** Enter name of Data

When the user enters the data name, ARDA will fit a model to the data and return with the answer. The user again is returned to the main menu. If the user decides to explore the data in more detail then RDADIS workspace should be copied by entering

**Command** )COPY Rdadis

This command copies all functions of workspace RDADIS into the current workspace.

**Response** The following menu appears

1 - TREND	NOT TESTED	TEST_LAPLACE
2- INDEPENDENCE	NOT TESTED	TEST_INDEPENDENC
3- PDF	NOT TESTED	TEST_PDF
4- LIKELIHOOD	NOT TESTED	TEST_LIKELIHOOD
5- LIFETIMES	NOT TESTED	TEST_LIFE
6- MAIN MENU		

To carry on any of these tests on data the user needs to enter the name of the driver function opposite to menu options. For example to test for independence the user should enter the following command

#### **Command Test\_Independenc**

This function will test the independence of two series X and Y which are vectors of numeric values.

**Response** ARDA will return the value 1 if X and Y were dependent, 0 if they were not dependent and 99 if it could not reach a result.

To fit a model to data the user should enter the command

**Command** Fit\_Dist\_All {A vector of names of distributions to be fitted}.

**Response** A two dimensional nested array of number of distributions each row will consist of two elements, the first element contains the parameters of distribution and the second element contains the loglikelihood of fitting the distribution to data. The distribution with the highest loglikelihood is fitted.

If data was not available and therefore not defined the user can use ARDA to obtain knowledge on the type of data that the user needs to achieve objectives. If the user required more Statistical knowledge or help on the job of different functions of ARDA then the workspace Rdaknow be copied and the user then will have a set of HOW functions to explain all how each function in ARDA operates.

As mentioned earlier, details on the different distributional models of data are contained in distributions vectors. These vectors each hold the name of one of the

distributions and in the current version of ARDA we have distribution vectors on **B2, FISK, GB1, GB2, LC, LN, and LT**. If the user desired to explore this knowledge, the user can enter

**Command**        **DIST**

**Response**    ARDA will list the name of all the distributional models available in ARDA and ask the user whether there are any more distributions to be added to ARDA. If the user responds positively the new distribution will be added to the existing list of distributions and then ARDA will ask for the name of distribution which the user desires to gain more knowledge on.

If the user required to interrogate the distributions vectors, for example to determine parents and children distributions of a particular distribution then,

**Command** **DISTCHILD {name of distribution}**

**Response**    The name of distributions immediately below this distribution in the Macdonald and Richards distribution tree (1987). If the user requires to know the parent distribution of a particular distribution then

**Command**        **PARENTS**

**Response**        name of parent distribution.

See Appendix 3 for further numerical examples.

## **7.6 Summary**

ARDA provides a model which could be used for a variety of types of data analysis. The system is innovatory in the use of objectives to guide analysis. To extend the system to other statistical analyses would require the definition of the set of objectives and models. It may also be necessary to modify object 'Data' to incorporate any peculiarities of the area being considered. A map of analysis would also have to be developed. The general framework would still be usable.

ARDA has been built partially using object oriented philosophy. Where appropriate this approach has been abandoned usually to avoid over elaboration or alternatively the ambiguity that can arise between message and object in OOP.

In this chapter we have described the system as used by an analyst to analyse data. There is, however, considerable information built into the system which will enhance the user's knowledge about Reliability Data Analysis. Each objective is linked to models and the models are linked to data. Hence using these links it will be possible for the user to explore which models are related to given objectives.

This inversion of the system arises out of APL's flexibility. Hence although we have described the system from the standpoint of analysis it can be used to carry out where necessary 'What-if' analyses.

There is no obvious impediment to generalising the systems approach to take account of the probabilities associated with hypotheses being tested and hence presenting the satisfaction of an objective in terms of probabilities. This could be taken further of course and a full Bayesian analysis could be implemented.

## **CHAPTER EIGHT**

### **FURTHER WORK**



## **8.1 INTRODUCTION**

The aim of this chapter is to review the achievements of the project and explore the further developments of the system. The first section will summaries the developments in the research. The second will suggest the further developments.

## **8.2 Summary**

The goals of this thesis were to:

- a) explore concepts within statistical expert systems  
and
- b) investigate the production of statistical expert systems.

To achieve these goals in this research it was decided to develop a system for reliability data analysis. It was primarily the intention of the research build a prototype which would demonstrate the feasibility of building the system. It was not the intention to build a commercial piece of software.

### **8.2.1 Concepts with Statistical Expert Systems**

In the light of these goals the project has had some success in the demonstrating how to build an expert system for statistical analysis. The original blue print for the system was GLIMPSE. Unfortunately GLIMPSE took a technique based approach rather than a problem lead approach. Also its statistical advisor was relatively inflexible. Having diverged from its advice the user was left to fend for their self.

Both of these problems have been tackled in the current system. A problem lead approach has been implemented. This has been achieved through the specification of the Objective which therefore acts as a control mechanism. To ensure this control mechanism can function it has been necessary to draw up an analysis map.

The map systemises reliability data analysis. It also provides a mechanism which need not be authoritarian. The control mechanism is then paths through the map. Having defined an objective the system will provide a path. the system will follow the map moving from one node to another. However, at each node the user will be able to decide on whether to follow the systems advice or decide to carry out an alternative strategy.

The second problem of inflexibility of the advice given by GLIMPSE has been overcome by the way knowledge is integrated into the system. Knowledge is located at the function level and at the node level of the map. There is also general knowledge available. By associating knowledge at the function, operational level, the user is still capable of being assisted when they have diverged from the system's advice. This is also true at the node level when applying specific techniques. The user still has the reassurance of advice on the technique and on the outcome.

The final implementation of the system could be compared with the statistical environment suggested by Neave (1989). There are operational level functions which the user can employ on a stand alone basis. On top of this there is the knowledge arising from a number of level, function, node and general. There is also guidance available on the steps of the analysis.

### **8.2.2 Implementation of Statistical Expert Systems**

The research has investigated a range of possible approaches to building a statistical expert system. Firstly it reviewed a number of possible statistical experts systems to locate a 'blue print'. Having decided on a knowledge enabling system there was a need to decide how this should be implemented. Initially the use of a expert system shell was considered. Such systems receive considerable criticism and in the current research this was equally found to be the case.

The next stage was to explore possible design philosophies that might be employed. Rule based and frame based systems were considered and they did

offer some desirable features. The need though to develop a system which had to be capability to be innovatory directed the search towards flexible design philosophies. Of such design philosophies object oriented programming seem to offer a range of advantages. This include the modularity of design.

The choice of programming language reflected both the need to develop statistical routines efficiently and the capability to conform to object oriented principles. Given APL2 could conform to these principles its advantages for statistical analysis made it a relevant choice. APL2 also provided the capability to deliver a statistical environment. Obviously if the project was started at the present time other languages or packages would be considered, for example S which has become more readily available.

### **8.3 Further Developments**

In this section a variety of practical issues are considered for the improvement of ARDA.

#### **8.3.1 ARDA's Front Ends**

Currently ARDA is constructed using APL2. It would now be possible to use APL2 windows version and so add the advantages of a windows which would improve the presentation. It would also desirable to introduce a graphical front end in ARDA. This can be a new major system object that will give ARDA the capability to represent data graphically. This object will communicate with other objects of the system via messages in the shape of a nested vector. Within the object GRAPHICS itself there might be specific processes to deal with specific paths of analysis, also the procedures to plot and display data with the facility to retain certain information on that particular job such as name of plotter or set up configuration. This conforms with the principles of Object - Oriented Programming.

### **8.3.2 Natural Language Development**

Currently ARDA contains several functions that define several logic relations such as IS\_IT , IS-THERE, NO and NOT. Obviously making use of developments within natural language would make ARDA more accessible and easier to use.

### **8.3.3 Interfaces**

To upgrade ARDA to commercial status it has to interface with and support other software packages such as Spreadsheets, databases, SPSS, S PLUS, BMDP, ASA and other packages. These interfaces will be useful producing input /output but should also introduce running code that will enhance ARDA's flexibility.

### **8.3.4 Data**

Data in general and in particular Reliability Data can be fairly problematic. It should be clearly defined with the inclusion of context. This being the case then most Reliability data is multivariate. In the current format of ARDA the DATA module caters for most types of data and enables the user to enter, define, and validate data. One area of improvement would be the development of interface facility to communicate with other Statistical systems, Spreadsheets and Databases. This could be useful especially in the process of data entry. Modern versions of APL allows the construction of interface facilities with other languages with ease. In APL2 this is achieved through Ancillary Processors and shared variables.

### **8.3.5 Models**

The main aim of this object is to select, define, and provide information on an appropriate model. These models will be composed of a Stochastic process and distributional form. Currently ARDA covers as far as Stochastic processes are

concerned Renewal Process, Non-Homogeneous Poisson Process and Poisson Process. It is possible to expand this section by including functions on Superimposed Renewal Process and Branching Process and some general Markovian models. As for distributional models in the current version of ARDA six data models have been used to represent the distribution of data these are Weibull, Gamma, Lognormal, Normal, Exponential, and Chi-Square. Further distributional forms can be included in ARDA to complement the six widely used Reliability distributions already built in, see McDonald & Richardad [1987].

Obviously it could be useful if we could interface with other packages such as SPSS, S-PLUS, ASA, BMDP and other packages of interest. The aim here could be to integrate much into the analysis map to avoid further development. This might be seen as using the object oriented nature of ARDA. One argument against this approach is that ARDA is no longer a stand alone system but requires presence of other software.

#### **8.3.6 Objectives**

In the current version of ARDA a limited set of Objectives is selected as a starting point. Further work could be carried on introducing new objectives to ARDA by defining the conditions required to satisfy these objectives and the tests needed. The Object - Oriented nature of ARDA enables it to expand its range of objectives without the need of major software change. It is also possible to organise objectives in a network that ease the analysis process by grouping achievable objectives together.

#### **8.3.7 Help and Guidance**

Currently there are HELP/GUIDANCE facilities available in ARDA, these could be improved by introducing a static set of sheets of knowledge on the different tests carried out by ARDA and possible analysis paths. Attempts to incorporate this facility in the current version of ARDA was unsuccessful due to screen control

problems caused by software and hardware restrictions. Otherwise it is relatively simple to implement through variables.

ARDA holds knowledge on data distributional models incorporated in distribution vectors. Currently these vectors partially cover the distributions tree by Macdonald and Richards (1987). It is possible to expand the set of distributions to contain all distributions.

It is also possible to improve assistance to user by developing menu driven topic selection facility of general Statistics and Data Analysis techniques to enable the statistically naive user update his/her knowledge on these topics. Another area of improvement is the design of a dynamic help function which will access any information required by the user during the analysis process. This could improve ARDA's flexibility and user friendliness. Although in the current version of ARDA help is available at each node of the analysis path map, it is possible to expand the help facility at these nodes by including menus of choices and suggestions.

#### **8.3.8 Explanation Facility**

In the current structure of ARDA it was designed to include an Error Trapping Facility by adopting a similar technique to that used in Help/Guidance functions. A specified number of lines in each function is reserved to detail the reasons behind possible error occurring when using these functions. ARDA also supports the user with the REPORT facility which contains knowledge of all the results of a session with ARDA.

General advice and support is available to user at main nodes of the system in the shape of general reasons describing the system decision to stop or choice of certain analysis path. A further stage in developing an explanation facility in ARDA is to create an explanation function in each system function we have which can be called whenever needed. It could be positioned in the same line at all functions and be called using a function called EXPLAIN.

## **REFERENCES**



## **REFERENCES**

- 1- K.P. Adlassnig, 1982 - A Survey of Medical diagnosis and Fuzzy subsets, In "Approximate Reasoning in Decision Analysis", M. Gupta and E. Sanchez, North Holland: Amsterdam, pp203-217.
- 2- M. Al-Doori, A Review of GLIMPSE, Professional Statistician Journal, 1990, Volume 9, Issue 10, pp 21-33.
- 3- M. Al-Doori, J.I. Ansell - Towards an Expert System, APL - Ication Conference proceedings, BCS, Kent, pp 65-72.
- 4- M. Al-Doori, J.I. Ansell - A volume for Reliability shelf - Reliability Data Analysis, British APL Association Journal (Vector), volume 9, number 2, 1991.
- 5- M. Al-Doori - A View Towards an Expert System for Reliability Data Analysis, Edinburgh University Working Papers Series, paper No. 933.
- 6- M. Alfonseca, 1988 - Object-Oriented Programming Introduction, IBM Madrid Scientific centre, Paseo de la Castellana, 4, 28043 Madrid, Spain.
- 7- M. Alfonseca,(1990), Object-Oriented Programming Tutorial, APL Quote Quad, 20, 7-8.
- 8- M. Alfonseca, 1990 - Neural Network in APL, APL90 Conference Proceedings, APL Quote Quad, Vol. 20, No. 4, pp2-6.
- 9- J.L. Alty and M.J. Coombs, 1984 - Expert Systems concepts and examples, NCC publication.
- 10- J.I. Ansell, 1987 - Msc. degree Information systems and expert systems - lecture notes, Department of Management Science, Hull University.
- 11- J.I. Ansell(1987) Analysis of lifetime data with covariates in proc. Natl Reliability Conference, Birmingham, Warrington, National Centre of System Reliability.
- 12- J.I. Ansell and M. Al-Doori, 1993 - ARDA - Expert System for Reliability Data Analysis, International Conference on APL, Toronto, Ontario, Canada, APL Quote Quad, vol. 24, No. 1.

- 13- R.O. Ansell and J.I. Ansell - Modelling the reliability of sodium sulphur cells, 1987, Reliab Eng, 17, 127-137.
- 14- J.I. Ansell, M.J. Phillips - Strategies for Reliability Data Analysis.
- 15- J.I. Ansell, M.J. Phillips - Some Problems with Graphical Methods to Assess System and Component Reliability, Working Paper Series NO. 93/4, Department of Business Studies, Edinburgh University.
- 16- J.I. Ansell, M.J. Phillips, 1989 - Practical Problems in the Statistical Analysis of Reliability Data (with discussion), RSS journal, series C, Volume 38, No.2, pp205 - 247.
- 17- J.I. Ansell, and M.J. Phillips, (1994), Practical Methods for Reliability Data Analysis', Oxford University Press: Oxford.
- 18- APL programming without tears, August 1993, APL Quote Quad, vol. 24, No.1, pp. 185-189.
- 19- APL 93 conference proceedings, APL Quote Quad, vol. 24, no. 1, August 1993.
- 20- H. Aplola, P. Peltola, 1990 - Integrating APL With symbol manipulation, numerical software and graphics, APL90 conference proceedings, APL Quote Quad, Vol. 20, No. 4, pp10-17.
- 21- H. Ascher, 1982 - Regression Analysis of repairable systems, reliability in proc. NATO Advanced Study Institute on Systems Effectiveness and Life cycle Costings, Springer, New York.
- 22- H. Ascher, H. Feingold - Repairable Systems Reliability - (1984) - Marcel Dekker - New York.
- 23- A.J. Ayer - 1986, "The Problem of Knowledge.", Pelican Books.
- 24- J. Banerjee, H. Chou, J.F Garza, W. Kim, D. Woelk, and N. Ballou, " Data Model Issues for Object-Oriented Applications," ACM Transactions on office Information Systems, vol. 5, no. 1, pp. 3-36, January 1987.

- 25- R.E. Barlow, R.E. Campo - Total time on test processes and application to failure data analysis. In Reliability and Fault Tree Analysis (e.d. R.E. Barlow, J.B. Fussel and N.D. Singpurwalla), pp451-481, Philadelphia, PA.
- 26- E.E. Bell - 1989, The development of an Expert System Approach of Experimental Data. Phd. Thesis, Mathematical Institute, University of Canterbury.
- 27- A. Bendell and D.M. Wightman, 1985 - The practical application of proportional hazards modelling. In proc. 5th Natl. Reliab. Conf. 2B/3/1-16. National Centre of system reliability. UKAEA, Warrington.
- 28- K.R. Carr, 1987 - A Framework for Selection of Expert Tools for IBM PC's, Data Systems Engineering Organisation, ISA, Paper #87-0516.
- 29- E. Castillo, E. Alvarez - Expert Systems: Uncertainty and Learning, Computational Mechanics Publications, Elsevier Applied Science.
- 30- P. R. Cohen- 1985, ' Heuristic Reasoning about                      Uncertainty : An Artificial Intelligence Approach, Boston : Pitmas.
- 31- A. Colmerauer, 1973 - Les Systemes-Q ou un Formalisme Pour Analyser et Synthetiser des phrases sur Ordinateur, Publication Interene No. 43, Dept. d'informatique, Universite de Montereal.
- 32- R.G. Cowell and A.P. Dawid- Applications of a general propagation algorithm for probabilistic expert system, 1991, Statistics and Computing 1992, 2, 25-36
- 33- B. Cox, Object-Oriented Programming : An Evolutionary Approach, Addison - Wesley publishing company, Reading Massachusetts, 1987.
- 34- D.R. Cox, PAW. Lewis, 1966 - The Statistical Analysis of Series of Events, Methuen, London.
- 35- D.R. Cox, D. Oakes, 1984 - Analysis of Survival Data, Methuen, London.
- 36- D.R. Cox, E.J. Snell, 1971 - Applied Statistics, Chapman Hall, London.
- 37- D.R. Cox- Regression models and life tables (with discussion), 1972, J.R. Stat. Soc. B, 34, 187-220.
- 38- D.R. Cox- Partial Likelihood. Biometrika, 1974 62, 269-276.

- 39- M.J. Crowder, A.C. Kimber, R.L. Smith and T.J. Sweeting, Statistical Analysis of Reliability Data, Chapman & Hall.
- 40- Crystal - user guide, Intelligent environments 1986.
- 41- R.O. Duda, J. Gasching and P.E. Hart, 1979 - Model design in PROSPECTOR consultant program for mineral exploration. In D. Michie (ed), Expert Systems in Microelectronic age, Edinburgh, Edinburgh University press.
- 42- E.A. Edmonds, L. Candy, P. Slatter and S. Lunn, 1989 - Issues in the Design of Expert Systems for Business, Expert Systems for Information Management, Vol. 2, No. 1, pp 1-22. - Crystal, [1986], User manual, Intelligent Environments Ltd., Griffin, London.
- 43- C. Farrands, 1990 - Expert systems: Growing Competition in Advanced Computing, EIU European Trends No. 4. -
- 44- Fox, J. [1985], 'Language, Logic and Uncertainty', Artificial Intelligence and Statistics, Addison-Wesley.
- 45- R. Forsyth, Expert Systems Principles and Case Studies, 1984.
- 36- Robert J. Frey, Object-Oriented Extensions to APL and J, Vector, 1992, vol. 9, no. 2, 116 - 125.
- 46- D.C. Gibson, G. Salvendy, 1990 - Knowledge representation in human problem solving implication for Expert System design, Behaviour & Information Technology, 1990, vol. 9, No. 3, pp 191 - 200. - L. Gilman, and A.J. Rose 'APL An Interactive Approach' Third Edition.
- 47- J. Girardot, 1990 - Arrays and References, APL90 Conference Proceedings, APL Quote Quad, Vol. 20, No. 4, pp 161-172.
- 48- A. Goldberg and D. Robson, 1983 - SAMLLTALK-80: The language and its implementation, Reading Mass, Addison-Wesley.
- 49- Gool, 1994 - The reality and future of expert systems, Information systems management, vol. VII, No. 1, pp 53-61.
- 50- M. Goldstein - 1989, The data trajectory, In Bayesian Statistics, 3, Oxford University Press.

- 51- I. J. Good, [1950], 'Probability and the Weighing of Evidence.', London : Griffin.
- 52- P.Hajek- Towards A Probabilistic analysis of MYCIN-like expert systems II, 1989, Expert Systems and Statistics workshop, Edinburgh University.
- 53- D.J. Hand - A Statistical Knowledge Enhancement System - J. R.Statist. Soc. A (1987) - 150 -part 4 - pp 334-345.
- 54- P. Hietala- Inside a statistical Expert System: Implementation of the ESTES system. In: Proc 20th symposium on the Interface (INTERFACE 88), ed. E.J. Wegma, D.T. Ganz and J.J. Miller (American Statistical Association, Alexandria, VA) pp 331-335.
- 55- IBM Research, 1992 - A Graphical Statistical System [AGSS] Reliability Analysis, IBM Research, Yorktown Heights, New York.
- 56- P. Jackson, 1985 - Reasoning about belief in the context of advice giving systems, (Ed, Bramer), pp. 73-83.
- 57- Peter Jackson, Introduction to Expert Systems - Second Edition,1991, McDonnell Douglas Research Laboratories, Saint Louis, Missouri, Addison-Wesley.
- 58- A.K.S. Jardine and M. Anderson, 1984 - Use of concomitant for reliability esatimation and setting component replacement policies. In Proc. 8th Adv. Reliab. Tech. Symp., Bradford, B3/2/1-6. National Centre of Systems Reliability, Ukaea, Warrington.
- 59- J.N.R Jeffers- A Seminar at Expert Systems and Statistics Workshop, Edinburgh University, 1989.
- 60- Judith Jeffcoate, Keith Hales and Valerie Downes: Object Oriented systems: The Commercial Benefits.
- 61- Won Kim & Frederick H. Lochovsky - Object-Oriented concepts, Database and Applications, 1989.
- 62- A.C. Kimber- Exploratory data analysis for possibly censored data from skewed distributions, 1990, Appl. Stat., 39, 21-30.

- 63- C.A. Kulikowski, [1980], ' Artificial Intelligence Methods and Systems for Medical Consultation 'IEE Trans Patt Anal Machine Intell., Pam-2464-475.
- 64- D. Lamberti, W.A. Wallace, 1987 - Presenting Uncertainty in Expert Systems: An Issue in Information Portrayal, Information & Management Journal, 13, pp159-169. -F. Land, T.Conford,1988 - In search of the Expert System Product, Expert Systems for Information Management, vol. 12, no. 2.
- 65- S.L. Lauritzen, Graphical chain models - Towards an expert system for the analysis of association, 1989, Research workshop, Expert Systems and Statistics, Edinburgh University.
- 66- S.L. Lauritzen and D.J. Spiegelhalter, [1988], ' Local Computations With Probabilities on Graphical Structures and Their Applications to Expert Systems ', Read to RSS January, 1988.
- 67- J.F. Lawless, 1982 - Statistical Models and Methods for Lifetime Data, J. Wiley & sons, New York.
- 68- J.F. Lawless, 1983 - Statistical Methods in Reliability (with discussion), Technometrics, vol. 25, pp 305-35.
- 69- J.F. Lawless, 1986 - A note on lifetime regression models, Biometrika, 73, 509-12.
- 70- D. Lenat, [1982], ' EURISKO : A Program That Learns the Heuristics and Domain Concepts. Artificial Intelligence, 21. v
- 71- PAW. Lewis, 1993 - A Graphical Statistical System [AGSS], APL 1993.
- 72- J. Liebowitz, L. C. Davis, W. F. Harris, V. H. Dean, 1989 - Expert Systems in Business and Information System Management: Developing CESA, Expert Systems for Information Management, Vol. 2, No. 1, pp 35-48.
- 73- D.V. Lindley, Scoring rules and the inevitability of probability, 1982, Int. Statist. Rev. 50, 1-26.
- 74- C.E. Lunneborg- LMG: A Knoeldge enhancement system for linear modelling, University of Washington press, Seattle 1990.

- 75- J. McDonald, D.O. Richards - Model Selection: Some Generalised distributions - Proc. Reliability '87 , Birmingham, 1987,, Culcheth.
- 76- J.A, MAdonald and J.Pedersen, Computing Environments for Data Analysis part 3: Programming environments, SIAM, J. for Sci & Stat. Comp.
- 77- N.R. Mann, R.E. Schafer and N.D. Singpurwalla, 1974 - Methods for the Statistical Analysis of Reliability and Life Data, J. Wiley & sons, New York.
- 78- B. Meyer, Object-Oriented Software construction, Prentice - Hall, Newyork 1988.
- 79- H.A. Mieg, 1993 - Computers on Experts : on the non-existence of expert systems, Psychology of decision and conflict series, Vol. 5.
- 80- D. Millet, 1992 - Expert systems success revisited, Discussion paper No. 92-54, Dept. of Accounting and Management Science, University of Southampton.
- 81- M. Minsky and S. Papert , 1969 - PRECEPTION: An Introduction to computational geometry. MIT press, Massachusetts.
- 82- MIL-HDBK-189, 1981 - Reliability Growth Management, HQ,US Army Communications Research and Development Command, ATTN: DRDCOPT, fort Monmouth, New Jersey.
- 83- M.F. Mohageg, 1991 - Object-Oriented versus bit-mapped graphics interfaces: Performance and preference differences for typical applications, Behaviour & Information Technology, Vol. 10, No. 2, PP 121-147.
- 84- Linda A. Murray, John T. E. Richardson - Intelligent Systems in a Human Context, Development, Implications and Applications, Oxford Science Publications.
- 85- P. Naeve, 1989 - Artificial Intelligence in Statistical Environments, Research Workshop, Expert Systems and Statistics, Held in Edinburgh June 28th - July 11th.
- 86- J.A. Nelder,1989 - GLIMPSE - Seminar at Dept. of Stat. - University of Edinburgh .



- 87- J. Nelder, 1990 - Talk given to RSS seminar on future of Statistical Computing. Royal Institution, London.
- 88- W. Nelson, 1993 - Accelerated life testing, John Wiley, New York.
- 89- M. Newby - Perspective on Weibull Proportional-Hazards Models, 1994, IEEE, Trans Reliability, R-43, 217-223.
- 90- O.M. Nierstrasz, " What is an 'Object' in object oriented programming?", in OBJECT AND THINGS, ed. D. Tschritzis, Universite De Geneve, Centre Universitaire D'Informatique, pp1-13, 1987. Reprinted from the proceedings of the CERN School of Computing, Renesse, The Netherlands, August 31-September 13, 1986.
- 91- P.D.T O'Connor- Practical Reliability Engineering, 1991, Heyden, London.
- 92- R.W. Olford- Statistical Data as Objects, 1989, Journal of Econometrics.
- 93- R.W. Olford- DINDE: Towards ....., 1989, SIAM J. of Stat. and Sci Computing.
- 94- G. Peper, C. MacIntyre and J. Keenan, 1990 - Hypertext: a New Approach for Implementing an Expert System, Expert Systems for Information Management, Vol. 3, No. 2, pp143-149.
- 95- C. Petrie, 1987 - Revised Dependency-Directed Backtracking for Default Reasoning, Proceedings of AI-87, Seattle.
- 96- S.Poltrock, D. Stiener and N. Tarlon, 1986 - Graphic Interfaces for Knowledge-Based System Development, Proceedings of ACM/SIGCHI, Boston, MA.
- 97- M. Minsky and S. Papert - PRECEPTION : An Introduction to computational Geometry, MIT Press Massachusetts, 1969.
- 98- Gerald E. Peterson - Tutorial: Object-Oriented Computing. Volume 1: Concepts.
- 99- Gerald E.Peterson - Tutorial: Object - Oriented computing. Volume 2 : Implementations, 1991.

- 100- Lewis J. Pinson and Richard S. Wiener, An introduction To Object Oriented Programming And SMALLTALK, Addison - Wesley, Reading Massachusetts, 1988.
- 101- U. Recker, M. RYS, 1990 - A Preferable Look - APL in Window Based environment, APL90 Conference Proceedings, APL Quote Quad, Vol. 20, No. 4, pp 312-321.
- 102- B.D. Ripley, 1992 - Statistical Aspects of Neural Networks, Lectures for SemStat, Sandbjerg, Denmark, Chapman & Hall.
- 103- F. Rosenblatt - The PRECEPTION : A Perceiving and Recognising Automation.
- 104- P. Roussel, 1975 - Groupe d'Intelligence Artificielle, Universite d'Aix-Marseille, Luminy.
- 105- J. Rumbaugh, Dec.1987 - Relations as a semantic constructs in an object oriented language, OOPSLA 87 conference proceedings, ACM SIGPLAN Notices, vol. 22, no. 12.
- 106- D. Russionoff, 1985 - A nonparametric inference system, MCC Technical report AI-062-85.
- 107- W. A. Rutiser, STSC - Object Oriented Programming of X window System Graphical User Interfaces, APL Quote Quad, 20, pp 331-341.
- 108-B.Sheil, 1984 - Power tools for programmers, Dtamation, Feb. pp 131-143.
- 109- E. Shortliffe and B.G. Buchanan - A model of Inexact Reasoning in Medicine, 1975, Math, Biosci., 23, 351-379.
- 110- N.D. Singpurwalla, 1988 - Foundational issues in Reliability and Risk Analysis, SIAM Review, 30, pp264-82.
- 111- Philip E. Slatter, Building Expert Systems - Cognitive emulation, Ellis Horwood Books in Information Technology.
- 112- D. Smellie, F. Evans, 1990 - Structured Expert system Design, APL90 Conference Proceeding, p362.
- 113- R.L. Smith, 1991 - Weibull regression models for reliability

- 114- P.J. Solomon, Effect of misspecification of regression models in the analysis of survival data, 1984, *Biometrika*, 71, 291-298.
- 115- D.J. Spiegelhalter, R.P. Knill-Jones - " Statistical and knowledge Based Approaches to Clinical Decision Support Systems with Applications in Gastroenterology , 1984 - *JRSS - B* - 146 - pp37-77.
- 116- D.J. Spiegelhalter - 1986, A Statistical View of Uncertainty in Expert Systems.
- 117- Spiegelhalter and Lauritzen- Sequential Updating of conditional Probabilities on directed graphical structures. Networks. 1990
- 118- Spiegelhalter , Franklin and Bull - Assessment, criticism and improvement of imprecise subjective probabilities in a medical expert system. Proceedings of 5th workshop on uncertainty in Artificial Intelligence.
- 119- George Springer & Daniel Friedman - Scheme and the art of programming.
- 120- G.L. Steele, Jr., Common Lisp: THE LANGUAGE, Digital Press, 1984.
- 121- B. Stroustrup, The C++ programming language, Addison-Wesley, Reading , MA, 1986.
- 122- Ernest R.Tello. Object - oriented programming for Artificial Intelligence, 1991.
- 123- J.W.B. Vermeulen, E.R.K. Spoor, 1990 - FRESH, An Expert System Design in APL2, APL90 Conference Proceedings, APL Quote Quad Vol. 20, No. 4, p391-403.
- 124- D.M. Weintraub, 1990 - APL2OS: Design Considerations for a Nested Array File System, APL90 Conference Proceedings, APL Quote Quad, Vol. 20. No. 4, pp 404-411.
- 125- R. S. Wiener, L. J. Pinson - An Introduction to Object-Oriented Programming and C++, 1991.
- 126- G.P. Zarri, 1988, Expert Systems for Information Management, In "Large Knowledge Bases", vol 1, No. 2.

127- D. Ziemann, 1992 - SMALLTALK, APL and J, An Exercise in Comparative Programming, British APL Association Journal (VECTOR), Vol. 9, No. 2, pp 96-15.

## **APPENDIX ONE**

### **USER GUIDE**

## **1 Introduction**

The aim of this appendix is to give the user the guidance required to start a session with ARDA. Each step of statistical analysis in ARDA is self explanatory and easy to follow.

## **2 Loading ARDA**

First of all you need to have a version of APL2 installed on your pc's hard disc and copy the system ARDA in its library. When starting APL2 you need to make sure what type of APL2 font is suitable for your PC. Most of IBM compatibles respond positively to the command APL2 FONT B which will load the APL2 system. To make sure that ARDA is available enter the command

)LIB RDA

a list of ARDA'S workspaces should be displayed on your monitor.

ARDA consists of seven workspaces, these are

**RDASTART.APL**

**RDADATA.APL**

**RDAMODEL.APL**

**RDAOBJEC.APL**

**RDADIST.APL**

**RDAKNOW.APL**

**RDAGUIDE.APL**

To start a session with ARDA you need to load the workspace **Rdastart.apl** by entering the command

**)LOAD RDASTART**

## **APPENDIX TWO**

### **THE QUESTIONNAIRE**



and to do that you will enter number 5. ARDA will ask you whether you want to save this session, if you do you need to enter the )SAVE command followed by the name of your current workspace you choose, for example let us call your current workspace AIRCRAFT99, the save command will be,

**)SAVE AIRCRAFT99**

All the functions and variables of the different workspaces you copied during you session with ARDA will be saved on your workspace. This enables you to resume your session with ARDA later if you desire to do so. All you need to do is to load your workspace is to enter the )LOAD command followed by the name of your workspace,

**)LOAD AIRCRAFT99**

To terminate your session with ARDA after saving you need to enter the command,

**)OFF**

## **1 Introduction**

At the early stages of this research it was decided to survey the use of expert systems in industry especially in Reliability. A list of possible expert systems users in industry was prepared totalling 250 names. Each user was sent a letter of introduction stating the reasons behind the survey and the main aims of the research. It also included a questionnaire consisting of a set of simple questions designed to establish the person's field of interest, attitude towards computers in general and expert systems in particular. A prepaid envelope was also included.

## **2 The Questionnaire**

The questionnaire consisted of sixteen questions divided into two main sections. The first section consisted of questions 1...8 which are general information questions, opposite to each is the score in percentage of the replies to each question. See section A3.2 for a summary of answers on question 1,2 and 3.

1- Name of Company / Institution and main area of work.

2- Your Name

3- Your role in company

4- Have you or are you currently using an expert system ? **YES 30% NO 70%**

5- If the answer to No 4 is NO are you contemplating the use of an expert system? If yes what system and for what use.

**YES 56.6% NO 43.3%**

6- If the answer is YES to 4 can you list the systems which you have or are currently using.

The majority of users were using the following systems, CRYSTAL, GLIMPSE, ART, BURNERS, SEAL, RISK, ENRICH, DARES, COMNET.

7- How much do you agree or disagree with each of the following statements. The response to this question can be one of six categories. These are,

strongly agree - agree - uncertain - disagree - strongly disagree - don't know

(a) I generally enjoy working with computers.

<b>Strongly agree</b>	<b>26.6%</b>
<b>Agree</b>	<b>63.3%</b>

(b) I believe expert systems are useful.

<b>Strongly agree</b>	<b>16%</b>
<b>Agree</b>	<b>37.5%</b>
<b>Uncertain</b>	<b>46.5%</b>

8- Have you any general comments to make about expert systems ?  
See section A3.3 a summary of comments made by respondents regarding their experience with expert systems.

The second section of the questionnaire consisted of questions 9...16 which are statements similar in style to question 7 and are designed to explore the user's opinion on types of expert systems he/she are familiar with and their features.

9- Would you please specify the field of application which your expert system, or systems, are used. [i.e. Medicine, Geology, Statistics, etc.]

10- The structure of the expert system you are using is,

<b>Rule Based</b>	<b>53%</b>
<b>Frame Based</b>	<b>41%</b>
<b>Both</b>	<b>0%</b>
<b>Other</b>	<b>6%</b>

If other please specify type.

11- How much do you agree or disagree with each of the following statements,

(a) The expert system you are using requires a large number of rules.

**Strongly agree**      **17.6%**

**Agree**                **52.9%**

**Uncertain**          **11.7%**

**Disagree**           **17.8%**

(b) The expert system you are using allows easy building of rules.

**Strongly agree**      **23.5%**

**Agree**                **52.9%**

**Uncertain**          **11.8%**

**Disagree**           **11.8%**

(c) The system is user friendly.

**Strongly agree**      **35.3%**

**Agree**                **35.3%**

**Uncertain**          **11.8%**

**Disagree**           **17.6%**

(d) The design of the front-end of the system is good.

**Strongly agree**      **11.8%**

**Agree**                **41.2%**

**Uncertain**          **35.3%**

**Disagree**           **5.9%**

**Strongly Disagree**   **5.8%**

(e) The system enables the user to easily update data.

<b>Strongly agree</b>	<b>5.9%</b>
<b>Agree</b>	<b>35.3%</b>
<b>Uncertain</b>	<b>17.6%</b>
<b>Disagree</b>	<b>30%</b>
<b>Strongly disagree</b>	<b>11.2%</b>

(f) The expert system you are using has explanation facilities.

<b>Strongly agree</b>	<b>5.9%</b>
<b>Uncertain</b>	<b>23.5%</b>
<b>Disagree</b>	<b>17.6%</b>
<b>Agree</b>	<b>53%</b>

(g) Explanation utilities are helpful to understand the system's results.

<b>Strongly agree</b>	<b>17.6%</b>
<b>Agree</b>	<b>35.3%</b>
<b>Uncertain</b>	<b>35.3%</b>
<b>Disagree</b>	<b>11.8%</b>

(h) The system gives accurate results.

<b>Strongly Agree</b>	<b>9.8%</b>
<b>Agree</b>	<b>54.6%</b>
<b>Uncertain</b>	<b>28.2%</b>
<b>Disagree</b>	<b>7.4%</b>

(I) The expert system you are using is cost effective.

<b>Agree</b>	<b>47.4%</b>
--------------	--------------

<b>Uncertain</b>	<b>21.4%</b>
<b>Disagree</b>	<b>23.7%</b>
<b>Don't Know</b>	<b>7.5%</b>

(j) Expert systems replace the expert in their field.

<b>Uncertain</b>	<b>58.8%</b>
<b>Disagree</b>	<b>29.4%</b>
<b>Strongly disagree</b>	<b>11.76%</b>

(k) More resources should be invested in your company on expert systems.

<b>Strongly Agree</b>	<b>11.8%</b>
<b>Agree</b>	<b>70.5%</b>
<b>Uncertain</b>	<b>17.7%</b>

12- When building an expert system what are the major problems?

<b>Building the knowledge base</b>	<b>11.76%</b>
<b>Constructing the rules</b>	<b>17.6%</b>
<b>The processing of data</b>	<b>23.5%</b>
<b>The system's output</b>	<b>29.4%</b>
<b>other</b>	<b>17.6%</b>

13- On a scale of 1 to 7 where 1 = good and 7 = poor, how would you rate the performance of the following functions of the expert system you are using,

	1	2	3	4	5	6	7
Presentation of System	23.5%	23.5%	35.4%	5.9%	11.7%		
Speed of System	17.6%	17.6%	53%	11.8%			
Building Knowledge	5.8%	11.7%	31.3%	31.3%	11.7%	5.8%	5.8%
Constructing Rules	17.6%	23.5%	47%	11.7%			
Degree of Accuracy	11.7%	35.5%	31.3%	23.5%			
Updating Knowledge base	11.7%	11.7%	17.6	23.5%	31.3%	5.9%	
Display of Results	31.3%	41.2%	23.5%				

14- Does the system you are using contain error traps ?

YES 62.5%

NO 37.5%

15- The training period on using expert systems at your company was,

Long enough	50%
Too short	31.3%
There was no training.	18.7%

16- If you have any more information regarding your experience with expert systems, please use the space below:

The response to the survey was 26% and of that 72% were expert system users. The answers of those with no experience in expert systems are in the table [1] and the answers of expert systems users are in table [2].

### **3 Respondents' Positions And Company Names**

- 1- Statistician -- Shell research ltd. ---- fuels & lubricants research.
- 2- Statistical consultant--- ICI fibers
- 3- Statistical services manger --- ICI Arochemicals--- pesticide manufacturing
- 4- Support for economists --- H.M. Treasury
- 5- Senior statisticians --- B.P. research
- 6- Mathematics & Statistics Consultant ----ICI chemicals & polymers
- 7- Statistics manager --- Glaxo Group Research --- Pharmaceutical research
- 8- Assistant consultant O.R. dept. -- British Steel, Central management services
- 9- Head of O.R. unit -- Dept. of Education & Science
- 10- Chief quality engineer -- Racal communications
- 11- System programmer -- Westland Helicopters



- 12- Senior research officer --- paint research associate
- 13- Business researcher --- British Gas
- 14- O.R. officer -- Property Services Agency
- 15- Computer System Manager --- Amersham International plc
- 16- Head of Statistics --- Dalgety plc. ---Group Research Lab. --Food manufacturing
- 17- Engineer - Lucas Engineering & systems --- Consultants/new techniques research
- 18- Ergonomist - VSEL -- Submarine Building
- 19- Senior Engineer -- W.S. Atkins -- Safety consultancy

#### **4 What Expert System , Why And Comments**

1- Expert systems are over hyped - should be thought of simply as research programs for solving simple diagnostic problems.

2- Trying to get a copy of DEXTER -- To aid the design of classical factorial experiments.--- Expert systems have the subject of rather too much hype. I have used "EXPERIMENTAL DESIGN" (1986) did no more than direct the user to literature references, which wasn't very helpful.

3-Within my organisation I do not believe that allocation of systems development effort to expert systems is justified. Also, I know of no package I wish to buy. I think I am saying that I believe expert systems are still in the research phase.

4- Sometimes wasteful in their disregard of other work - e.g. in statistics or in database query languages.

5- Might use GLIMPSE if I ever required GLIM --- Statistical Analysis

6- I would use expert systems as front end to statistics packages, especially with respect to experimental design --- Expert systems are solutions looking for

problems to solve. My impression is that such systems are be-spoke. Are there any generic tools which can be used to create useful systems?

7- GLIMPSE----Statistical analysis --- I believe that expert systems currently available in the statistics area are really aids for statisticians not for others. I do have concerns about their use in the hands of the inexperienced.

8-CRYSTAL -- we do not use expert systems currently but we build them for other parts of British Steel. -- Credit assessment.

9- CRYSTAL & DbaseIII

10- Ferranti ART --- Military Battlefield scenarios --- Decision options to commander.

My role is one of quality assessment - this is relatively easy (but expensive & time consuming) for traditional software including expert systems but to validate the requirement for expert systems (i.e. the knowledge model) - UGH!!. They can only be used as general aids - certainly until there is a scientifically verifiable means of acquiring knowledge. In case you are concerned at the use of expert systems in military applications - they are still in the evaluation stage. The associated relational DB's need to be enormous, dynamic etc, and the knowledge base heavily orientated towards "human" factors - expert systems are easy to implement for Medicine or other fault diagnostic - less so for "thought processes" . Please advise if you have an effective method of assessing "quality" for this type of expert system.

11-CRYSTAL - Outside developed rulebased training system

12- See DTI report on expert system and usage. The lack of economic benefit certainly seems to be true in our case.

13- BURNERS- design of industrial burners

SEAL - advice / solution for sealing gas leaks. Currently using

RISK - risk analysis of damage to computer centres

ENRICH - noisy central heating diagnostic

DARES - Interpretation of management accounts

COMNET - problem solving in computer network.

14- Selection of appropriate building items - prices for specification x price book matrix to synthesise appraisal of tender value.

We co-operated in general development of system outlined in Q5 as part of the ALVEY initiative. The task appeared relatively simple and ideally suited to an expert system. Eventually we had to conclude that it was not cost effective in terms of resources - the pay back period will be infinite. It seems to me that relationship between sophistication and resource is exponential or worse. Our need for precision put it out of our reach.

## **5 Conclusion**

The questionnaire was useful in that it gave us a general idea about the number of software users in industry involved with expert systems and reflected their opinion and some of the problems facing them with expert systems. The answers of the respondents seem to point at two main problems associated with using expert systems. The first problem the high cost of developing expert systems and the difficulty to justify that cost to management.

The second problem is the low level of accuracy of results given by some expert systems makes it difficult for users to depend on them. This indicates the need for further research into expert systems to produce better systems.

The following is a copy of the questionnaire.

Survey on Expert Systems in use :-

The aim of this questionnaire is to survey the current usage of Expert Systems and attempt to identify the major concerns in using such systems. We hope you will be willing to spend a little time answering the following. Your co-operation will be appreciated and answers will be treated in strictest confidence.

1- Name of Company / Institution and main area of work.

2- Your name .....

3- Your role in company .....

4- Have you or are you currently using an Expert System ?

YES		NO	
-----	--	----	--

5- If the answer is NO to ( 3 ) are you contemplating the use of an Expert System ?

YES		NO	
-----	--	----	--

If YES what system....

for what use....

6- If the answer is YES to ( 4 ) can you list the systems which you have or are currently using .

Have used....

Currently using....

7- How much do you agree or disagree with each of the following statements :

( a ) " I generally enjoy working with computers. "

Strongly agree	Agree	Uncertain	Disagree	Strongly disagree	Don't Know

( b ) " I believe Expert Systems are useful. "

Strongly agree	Agree	Uncertain	Disagree	Strongly disagree	Don't Know

8- Have you any general comments to make about Expert Systems ?

\*\*\*\*\*  
\* For those who have not experienced Expert Systems \*  
\* thank you for your co-operation in answering this \*  
\* questionnaire. \*  
\* \*  
\*\*\*\*\*

9- Would you please specify the field of application which your Expert System, or systems, are used. [ i.e. Medicine, Geology, Statistics, etc.]

10- The structure of the Expert System you are using is :

Rule based		Frame based		both		Other	
------------	--	-------------	--	------	--	-------	--

If other please specify type

11- How much do you agree or disagree with each of the following statements ?

( a ) " The Expert System you are using requires a large number of rules. "

Strongly agree	Agree	Uncertain	Disagree	Strongly disagree	Don't Know

( b ) " The Expert System you are using allows easy building of rules. "

Strongly agree	Agree	Uncertain	Disagree	Strongly disagree	Don't Know

( c ) " The system is user friendly. "

Strongly agree	Agree	Uncertain	Disagree	Strongly disagree	Don't Know



( d ) " The design of the front-end of the system is good. "

Strongly agree	Agree	Uncertain	Disagree	Strongly disagree	Don't Know

( e ) " The system enables the user to easily update data. "

Strongly agree	Agree	Uncertain	Disagree	Strongly disagree	Don't Know

( f ) " The Expert System you are using has explanation facilities. "

Strongly agree	Agree	Uncertain	Disagree	Strongly disagree	Don't Know

( g ) " Explanation utilities are helpful to understand the system's results. "

Strongly agree	Agree	Uncertain	Disagree	Strongly disagree	Don't Know

( h ) " The system gives accurate results. "

Strongly agree	Agree	Uncertain	Disagree	Strongly disagree	Don't Know



( i ) " The Expert System you are using is cost effective. "

Strongly agree	Agree	Uncertain	Disagree	Strongly disagree	Don't Know

( j ) " Expert Systems replace the expert in their fields. "

Strongly agree	Agree	Uncertain	Disagree	Strongly disagree	Don't Know

( k ) " More resources should be invested in your company on Expert Systems. "

Strongly agree	Agree	Uncertain	Disagree	Strongly disagree	Don't Know

12- When building an Expert System what are the major problems?

Building the knowledge base	
Constructing rules	
The processing of data	
The system's output	
Others	

If others please specify :

13- On a scale of 1 to 7, where 1 = good and 7 = poor, how would you rate the performance of the following functions of the Expert System you are using.

Presentation of system							
Speed of system							
Building knowledge base							
Constructing Rules							
Degree of accuracy							
Updating knowledge base							
Display of results							

14- Does the system you are using contain error traps ?

YES		NO	
-----	--	----	--

15- The training period on using Expert Systems at your company was :

Too long	
Long enough	
Too short	
There was no training	

16- If you have any more information regarding your experience with Expert Systems, please add it below:

\*\*\*\*\*  
\*  
\*  
\*        THANK YOU    FOR YOUR CO-OPERATION AND HELP        \*  
\*  
\*  
\*\*\*\*\*

### **APPENDIX THREE**

#### **LISTING OF SYSTEM BUILT IN CRYSTAL**

## **APPENDIX THREE**

### **NUMERICAL EXAMPLES**

## **NUMERICAL EXAMPLES**

The following are number of example functions in ARDA and some details on how the system is used.

To start a session with ARDA the user needs first to load the APL2 software by entering APL2 directory in the C: drive and entering the command APL2. The user needs to load the system ARDA by entering the command

**)LOAD ARDA**

the following message will be displayed

**AN EXPERT SYSTEM FOR RELIABILITY DATA ANALYSIS.**

**COPYRIGHT JAKE ANSELL & MULHIM AL-DOORI.**

**PLEASE ENTER THE COMMAND START TO BEGIN A NEW SESSION WITH ARDA.**

**START**

**DO YOU WISH TO START A NEW SESSION ? Y/N**

If the user wants to use a previous workspace on the system the answer entered should be NO.

If the user wants a fresh workspace with ARDA the answer should be YES. The system will return the following message,

**CHOOSE ONE OF THE FOLLOWING OPTIONS.**

- |                                 |                   |
|---------------------------------|-------------------|
| <b>1- DATA ANALYSIS</b>         | <b>NO</b>         |
| <b>2- OBJECTIVES ANALYSIS</b>   | <b>NOT TESTED</b> |
| <b>3- MODELS ANALYSIS</b>       | <b>NOT TESTED</b> |
| <b>4- STATISTICAL KNOWLEDGE</b> | <b>NOT TESTED</b> |
| <b>5- QUIT</b>                  | <b>NOT TESTED</b> |

**ENTER THE NUMBER OF ONE OF OPTIONS.**

Let us assume that the user is starting from scratch and will choose to analyse data, so option one is chosen y entering number 1, the system will return the following message,

**YOUR CHOICE IS 1 PLEASE CONFIRM Y/N**

This is to double check whether the user entered the correct number, so the user should confirm the choice by entering Y. What follows is a set of questions asked by ARDA to determine the nature of data. In each question the possible answers are put in brackets.

**IS THE DATA A NEW SET (NEW) OR AN OLD SET (OLD) ?**

If the user needs to further analyse, add details to or update an already entered set of data then the correct answer to enter is OLD, otherwise if the user intends to enter a new set of data the choice is NEW. ARDA will then interrogate the user on details of the new set of data. So let us assume it is a new set of data,

NEW

**IS THE DATA LIFETIME (LIFE) OR AGGREGATE DATA (AGG), I DO NOT KNOW (NOT) ?**

The user in these questions is given the DO NOT KNOW option to cover lack of knowledge and the possibility of adding more information at a later stage or another session. The data we are using in this set of examples is a set of lifetimes so we should enter,

LIFE

**IS DATA A SINGLE COMPONENT DATA (COMPONENT) OR IS DATA A SEQUENCE DATA (SEQUENCE) ?**

COMPONENT

**IS DATA A SINGLE SEQUENCE (SINGLE) OR A MULTIPLE SEQUENCE (MULTIPLE) ?**

SINGLE

**IS DATA A COVARIATE DATA (COVARIATE) OR IS DATA A NO COVARIATE DATA (NO) ?**

NO

**IS DATA CENSORED DATA (CENSORED) OR IS DATA NOT CENSORED DATA (NOT) ?**

NOT

**WHAT IS THE DATA COLLECTED ON ?**

AIRCRAFT COMPONENTS

**WHAT ARE THE COMPONENTS OF DATA ?**

LIFETIMES OF AIRCRAFT TYRES

**WHAT IS THE SERIAL NUMBER ?**

17-7-1968



**WHERE WAS THE DATA COLLECTED ?**

**EDINBURGH**

**DOES DATA HAVE ANY PHYSICAL CONTEXT ?**

**NO**

**ARE THERE ANY PECULIARITIES IN DATA ?**

**NO**

**ENTER DATA**

The user is now expected to enter data. The data we are using is a set of lifetimes we shall call  
EXAMPLE, It is of the following format,

5 10 17 32 32 33 34 36 54 55 55 58 58 61 64 65 65 66 67 68 82 85 90 92 92  
102 103 106 107 114 114 116 117 124 139 142 143 151 158 195.

**ENTER NAME OF DATA**

**EXAMPLE**

This is the end of the data analysis option. The information entered above will be stored in a  
nested array vector called DATA which will form part of the ARDA's main nested vector  
REPORT. The system then returns the following message with the main menu,

**THE FOLLOWING IS A SET OF OPTIONS AVAILABLE IN THE SYSTEM TO  
USE.**

**YOU CAN DECIDE ON THE PATH OF ANALYSIS YOU REQUIRE BY TYPING  
THE OPTION NUMBER OR F AVAILABLE THE NAME OF THE FUNCTION  
SHOWN NEXT TO YOUR SELECTED ACTIVITY.**

**CHOOSE ONE OF THE FOLLOWING OPTIONS.**

<b>1- DATA ANALYSIS</b>	<b>YES</b>
<b>2- OBJECTIVE ANALYSIS</b>	<b>NOT TESTED</b>
<b>3- MODEL ANALYSIS</b>	<b>NOT TESTED</b>
<b>4- STATISTICAL KNOWLEDGE</b>	<b>NOT TESTED</b>
<b>5- QUIT</b>	<b>NOT TESTED</b>

Note the word YES declared next to the first option changing the previous statement of NOT  
TESTED declaring that it has been selected previously. Having entered data let us assume that

the user will desire to select the second option of OBJECTIVE ANALYSIS. To do so the user will enter the number of the option,

2

**YOUR CHOICE IS 2 PLEASE CONFIRM Y/N**

Y

**HAVE YOU ANY OBJECTIVES OR ARE YOU EXPLORING THE SYSTEM ?  
OBJ/EXPL**

If the user is unaware of the set of objectives achievable by the system then the EXPLORE choice can be selected, the following message will be displayed,

**TO EXPLORE THE DATA VECTOR PLEASE ENTER DATA**

**TO EXPLORE THE MODEL VECTOR PLEASE ENTER MODEL**

**TO EXPLORE THE OBJECTIVE VECTOR PLEASE ENTER OBJECTIVE.**

**TO ENQUIRE ABOUT ANY FUNCTION IN ARDA PLEASE ENTER**

**^HOW^FUNCTION *NAME OF FUNCTION***

**TO EXPLORE LIST OF REPORTS AVAILABLE IN ARDA PLEASE ENTER**

**REPORT**

**TO PRINT ANY OF THESE REPORTS PLEASE ENTER**

**PRINTFUNCTION '*REPORT NAME*'**

If the user desires to go ahead with attempting to achieve an objective, the second option is selected and the following command should be entered,

**OBJECTIVE**

The system will return the following message.

**PLEASE ENTER YOUR CHOICE NUMBER,**

<b>1- TREND</b>	<b>NOT TESTED</b>
<b>2- INDEPENDENCE</b>	<b>NOT TESTED</b>
<b>3- PDF</b>	<b>NOT TESTED</b>
<b>4- LIKELIHOOD</b>	<b>NOT TESTED</b>
<b>5- LIFETIMES</b>	<b>NOT TESTED</b>
<b>6- CALCULATE STATISTICS</b>	<b>NOT TESTED</b>
<b>6- MAIN MENU</b>	

## **1 TEST TREND**

This objective will test whether there is a trend in data or not and what type of trend is it. It uses the LAPLACE test. The user can either use this function by choosing OBJECTIVES option in the main menu and then selecting option number 1 in the objectives menu, or by independently entering the following command using the above set of data (EXAMPLE):

### **TEST\_LAPLACE EXAMPLE**

The system will return the following message,

**1**

**-1.964207618**

The figure 1 indicates a successful check for trend followed by the value of the LAPLACE trend test. These two figures are followed by the following statement,

**“WHAT LEVEL OF SIGNIFICANCE DO YOU WISH TO TEST AT, THE LEVEL SHOULD BE EXPRESSED AS A PERCENTAGE,**

if the user enters the level as 0.05 the system returns the following result

**-1.964207618**

**TREND IMPROVING**

**THE FOLLOWING IS A SET OF OPTIONS AVAILABLE TO CHOOSE FROM; TO CHOOSE AN OPTION PLEASE TYPE THE COMMAND IN [ ] BRACKETS AND PRESS RETURN.**

- |   |                      |
|---|----------------------|
| <b>1- NHPP</b>  | <b>[NHPPFIT]</b>     |
| <b>2-OTHER NON-STATIONARY MODEL, E.G. CROWS MODEL</b> |                      |
| <b>[NONSTATMOD]</b>                                   |                      |
| <b>3- MAIN MENU</b>                                   | <b>[START]</b>       |
| <b>4- ANY OTHER ACTIVITY THE USER DESIRES</b>         | <b>[USER DESIRE]</b> |
| <b>5- QUIT</b>  | <b>[END]</b>         |

The user will then choose the appropriate path of analysis.

**2 TEST INDEPENDENCE**

This objective will test whether data is dependent or independent. This function is used either by selecting the OBJECTIVES option on the main menu and then choosing option 2 in the objectives menu, or by entering the following command, using the same set of data (EXAMPLE),

**TEST\_INDEPENDENCE EXAMPLE**

The system will return the following result

**0.04368529281**

**DATA IS INDEPENDENT**

The result consists of the autocorrelation coefficient and a message indicating whether data is dependent or independent. This message is followed by the main menu.

**3 TEST DATA'S STATISTICS**

This objective is useful to assess the performance of a single population against a fixed rate or to compare several populations. There are several functions available in ARDA to analyse data, for instance and using the same data set EXAMPLE, the user can enter the following functions to calculate data statistics,

<b>MEAN</b>	<b>(TO CALCULATE MEAN)</b>	<b>= 55.04</b>
<b>SDEV</b>	<b>(TO CALCULATE STANDARD DEVIATION)</b>	<b>= 24.72764984</b>
<b>VAR</b>	<b>(TO CALCULATE VARIANCE)</b>	<b>= 611.4566667</b>
<b>CV</b>	<b>(TO CALCULATE COEFFICIENT OF VARIANCE)</b>	<b>= 0.4492668939</b>
<b>PCTILE</b>	<b>(TO CALCULATE DATA'S QUANTILES)</b>	<b>25 =34</b>
		<b>50 =58</b>
		<b>75 =67</b>
<b>MEDIAN</b>	<b>(TO CALCULATE DATA'S MEDIAN)</b>	<b>=58</b>

The system will then return to the main menu with NOT TESTED message opposite to option 2 changed to TESTED declaring that this option has been chosen during this session. The user then will select option number 3 which is MODELS ANALYSIS by entering the number 3,

this option will invoke the function COMP\_PDF which is designed to compare the results of different likelihood tests to choose the best of them. ARDA in its current form caters for a limited number of distributions, these are WEIBULL, NORMAL GAMMA, LOGNORMAL and EXPONENTIAL. The functions WEIBPDF, NORMPDF, GAMMAPDF, LOGNORMPDF and EXPOPDF are designed to calculate the probability density function for their prospective distributions. The functions WEIBEST2, NORMEST, GAMMEST, LOGNEST and EXPOEST are used to estimate the parameters for each distribution. All these functions are called within the function COMP\_PDF to select the best fit for the set of data being analysed. The following is a description of two of these functions WEIBEST” and GAMMEST.

### 1 WEIBEST2

$$X\sim WEI(\theta,\beta)$$

The WEIBEST2 function is discussed here as a numerical example of the system’s functions. The aim of this function is to provide estimation technique for the weibull distribution or (extreme value distribution). The Weibull distribution [MA1]has been used successfully to describe failure times. the conditional failure rate for the Weibull model is given by:

$$Z(t) = m\lambda^{m-1}$$

This function depends on two parameters, m and λ, which are called the Weibull shape parameter and the weibull scale parameter respectively. For the special case when m = 1, we find Z(t) = λ, so the exponential model is a special case of the weibull model. If the weibull distribution is thought to be appropriate, then both Weibull parameters must be estimated from the data.

The lifetimes data set EXAMPLE is to estimate the Weibull parameters,

Command	<b>WEIBEST2 EXAMPLE</b>	
Result:	<b>0.0001490642209</b>	<b>1.944882346</b>

**2 GAMMEST**

$x \sim \text{GGD}(\vartheta, \beta, \kappa)$

The aim of this function is to estimate the parameters of the Gamma distribution. The Gamma distribution is used as a lifetime model, though not nearly as much as the Weibull distribution. This is partly because the survivor function and hazard function of the Gamma are not expressible in a simple closed form.

$$I(k, \lambda t) = \frac{1}{\Gamma(k)} \int_0^{\lambda t} x^{k-1} e^{-x} dx$$

Using the above set of data example

**GAMMEST EXAMPLE**

**23.241694      0.001851247317**

Using the sample mean and sample variance.

The estimated parameters for the other distributions for the same data set EXAMPLE are,

<b>NORMAL</b>	1921.507051	82.675
<b>LOGNORMAL</b>	8.39776573	4.216080197
<b>EXPONENTIAL</b>	82.675	

Going back to our session with ARDA the user now may choose to select option 3 which is MODEL ANALYSIS to fit a model to data.

3  
**YOUR CHOICE IS 3 PLEASE CONFIRM Y/N**  
Y  
**ENTER NAME OF DATA**  
EXAMPLE  
**GAMMA DISTRIBUTION**

ARDA has fitted the GAMMA distribution to the data set EXAMPLE.

The system will then return to the main menu declaring option 3 has been tested. The user can then select option number 4 if he/she desires so by entering number 4,

4

YOUR CHOICE IS 4 PLEASE CONFIRM Y/N

Y

THE AIM OF THIS FUNCTION IS TO PRODUCE A POOL OF INFORMATION ON SUBJECTS RELATED TO ARDA AND TO RELIABILITY DATA ANALYSIS. TO ACCESS KNOWLEDGE ON ANY OF THE TOPICS JUST ENTER ITS NUMBER AND PRESS RETURN,

- |                      |                    |                     |
|----------------------|--------------------|---------------------|
| 1- OBJECTIVES        | 2- MODELS          | 3- DATA             |
| 4- NORMAL DIST.      | 5- GAMMA DIST.     | 6- WEIBULL DIST.    |
| 7- EXPONENTIAL DIST. | 8- LOGNORMAL DIST. | 9- NHPP             |
| 10- BPP              | 11- HPP            | 12- RENEWAL PROCESS |
| 13- IHPP             | 14- B2             | 15- GB1             |
| 16- GB2              | 17- GG             |                     |

Each one of these options holds information related to its topic, for example DATA is the nested vector that contains all the details of the DATA ANALYSIS phase of ARDA. If we enter DATA then the following report is displayed by ARDA

DATA IS DEFINED  
NEW SET OF DATA  
LIFETIMES  
SINGLE COMPONENT  
NO COVARIATES  
DATA COLLECTED ON AIRCRAFT COMPONENTS  
AIRCRAFT TYRES LIFETIMES  
SERIAL NO. 17-7-1968  
DATA HAS NO PHYSICAL CONTEXT  
DATA HAS NO PECULIARITIES  
DATA IS        5 10 17 32 32 33 34 36 54 55 55 58 58 61 64 65 65 66 67 68 82 85 90 92 92  
                 102 103 106 107 114 114 116 117 124 139 142 143 151 158 195  
DATA'S NAME IS (EXAMPLE)



Other options such as **MODEL** is the nested vector which contains details whether the model is fitted or not to data and what type of process is it stochastic or distributional, and type of model fitted. If we enter the option number 2 ARDA will display the following message,

**MODEL DEFINED                      GAMMA**

ARDA also contains knowledge on a set of distributions described in McDonald & Richards (1987). The information is displayed by ARDA in the following format,

**NAME                      PARAMETERS                      PARENTS                      CHILDREN                      ESTIMATION**

that is the distribution's name, its parameters, its parents and children in the McDonald & Richard distribution tree, the function used to estimate its parameters in ARDA and finally the function used to calculate its probability density function.

For example if the user requires to gain knowledge on the Exponential distribution, option 7 can be selected and the following message will be displayed,

<b>EXP</b>	<b>b</b>	<b>a=1++</b>	<b>w</b>	<b>EXPOEST</b>	<b>EXPOPDF</b>
		<b>p=1</b>	<b>GAMMA</b>		
		<b>q=a*</b>	<b>L</b>		

Another example can be selecting the GENERALISED GAMMA distribution by entering option number 17. ARDA will display the following information,

<b>GG</b>	<b>aqp</b>	<b>q-x*</b>	<b>GB1</b>	<b>q-O***</b>	<b>GAMMEST</b>	<b>GAMMAPDF</b>
		<b>q++</b>	<b>GB2</b>	<b>p=1w</b>		
				<b>a=1++GA</b>		
				<b>a=2 NORMAL</b>		
				<b>p=1/2</b>		

At the end of exploring knowledge in ARDA the user can return to main menu and choose to carry on the analysis. The user can call each function in ARDA individually and use it or retrieve information on it available in a pool of HOW functions as mentioned earlier. When the user needs to end the session with ARDA option 5 is chosen from the main menu which is the

QUIT option. The user is then given a choice to save the workspace where he/she worked if so the user should enter the command

**)SAVE**

otherwise a session with ARDA is terminated by entering the command

**)OFF**

## **APPENDIX FOUR**

The Following is a listing of the knowledge based built using CRYSTAL as a version of ARDA in a hierarchical structure.

```

[ 23]          + IF [ 9] INPUT
              + AND [ 4] DISPLAY DATA

[ 9] INPUT
  IF
    DO: Menu Question dimension

      Is the array a
        Single dimension array { }
        Double dimension array { }
        COL : SURR           White on Blue
        COL : 0,0            White on Blue
        MENU: 4,28,38        Blue on Gray      Lt_Red
        MENU: 6,28,38        Blue on Gray      Lt_Red

        AND DO: Test Expression
              dimension=1
        + AND [ 10] INPUT1

        OR DO: Test Expression
            dimension=2
        + AND [ 12] INPUT2

[ 10] INPUT1
  IF
    DO: Display Form
      | Enter the name of array <name$ >
      How many variables are you entering ?
      <totalnumber >
      COL : SURR           White on Blue
      COL : 0,0            White on Blue
      IN  : 0,30,49        Blue on Gray
      name$
      IN  : 4,11,25,0      Blue on Gray
      totalnumber

      + AND [ 14] INPUTa

[ 14] INPUTa
  IF
    DO: Assign Variable
      number:=number+1
    AND DO: Display Form
          [name$ ]
          Enter new value <new_v>
          number is [number]
  
```

total number is [totalnu]

[new\_val]

COL : SURR           White on Blue  
COL : 0,0            White on Blue  
OUT : 0,0,20  
      name\$  
OUT : 4,20,27,0  
      number  
OUT : 6,26,34,0  
      totalnumber  
OUT : 8,11,19,2  
      new\_value  
IN : 2,17,23,2       Blue on Gray  
      new\_value

AND       DO: Assign Variable  
          obs[number]:=new\_value  
AND       DO: Test Expression  
          number<totalnumber  
AND       DO: Restart Rule

OR        DO: Assign Variable  
          number:=0

[ 12] INPUT2  
      IF

DO: Display Form  
   Enter double array name     <name\$                   >  
  
   Enter the number of columns <totalnumber           >  
  
   Enter the number of rows    <totalindex            >  
COL : SURR            White on Blue  
COL : 0,0            White on Blue  
IN : 0,27,47        Blue on Gray  
      name\$  
IN : 2,29,46,0       Blue on Gray  
      totalnumber  
IN : 4,29,45,0       Blue on Gray  
      totalindex

+   AND [ 15] INPUTb

[ 15] INPUTb  
      IF

AND       DO: Assign Variable  
          number:=0  
          DO: Menu Question   waydat  
  
          How would you like to input the array  
  
          row by row           {                   }  
  
          column by column     {                   }  
COL : SURR            White on Blue

\*  
\* RULE LIST  
\* -----

Mon Aug 14 06:46:39 1995 Page: 3

	COL : 0,0	White on Blue	
n Blue	MENU: 3,26,35	Blue on Gray	Lt_Red
n Blue	MENU: 5,26,35	Blue on Gray	Lt_Red

```

      AND      DO: Test Expression
                waydat=1
      AND      DO: Assign Variable
                index:=0
+   AND [ 11] inputlb
      AND      DO: Restart Rule

      OR      DO: Test Expression
                waydat=2
      AND      DO: Assign Variable
                number:=0
+   AND [ 13] INPUT2b
      AND      DO: Restart Rule

```

```

[ 11] inputlb
+ IF [ 16] INPUTb1a
  AND DO: Assign Variable
        number:=0
  AND DO: Assign Variable
        index:=index+1
  AND DO: Test Expression
        index<totalindex
  AND DO: Restart Rule

OR DO: Assign Variable
   index:=0

```

```

[ 16] INPUTb1a
IF DO: Assign Variable
   number:=number+1
AND DO: Display Form
      [name$

```

```

      Enter the new value <ad_value >

      The total number of columns is [totalnumber]

      The total number of rows is [totalindex ]

      The new value is [new_value ]
      number [nu] index [in]
COL : SURR      White on Blue
COL : 0,0      White on Blue
OUT : 0,13,32
      name$
OUT : 4,34,46,0
      totalnumber
OUT : 6,31,44,0
      totalindex
OUT : 8,24,38,0

```

```

new_value
OUT : 9,7,10,0
number
OUT : 9,19,22,0
index
IN : 2,22,33,0   Blue on Gray
ad_value

AND DO: Assign Variable
arr[number,index]:=ad_value
AND DO: Test Expression
number<totalnumber
AND DO: Restart Rule

OR DO: Assign Variable
number:=0

[ 13] INPUT2b
+ IF [ 17] INPUTb2a
AND DO: Assign Variable
number:=0
AND DO: Assign Variable
index:=index+1
AND DO: Test Expression
number<totalnumber
AND DO: Restart Rule

OR DO: Assign Variable
number:=0

[ 17] INPUTb2a
IF DO: Assign Variable
number:=number+1
AND DO: Display Form
[name$

Enter the new value <ad_value >

The tortal number of columns is [totalnum]

The total number of rows is [totalindex ]
The new value is [new_value ]

number [numbe] index [index ]
COL : SURR White on Blue
COL : 0,0 White on Blue
OUT : 0,11,36
name$
OUT : 4,34,43,0
totalnumber
OUT : 6,32,44,0
totalindex
OUT : 7,21,33,0
new_value

```



OUT : 9,9,15,0  
 number  
 OUT : 9,33,41,0  
 index  
 IN : 2,22,34,0 Blue on Gray  
 ad\_value

AND DO: Assign Variable  
 arr[number,index]:=ad\_value  
 AND DO: Test Expression  
 number<totalnumber  
 AND DO: Restart Rule  
 OR DO: Assign Variable  
 number:=0

[ 4] DISPLAY DATA  
 IF

DO: Menu Question data  
 What would you like to display ?

A single line array { }

A whole array { }

Single array numbers { }

COL : SURR White on Blue  
 COL : 0,0 White on Blue  
 MENU: 2,32,41 Blue on Gray Lt\_Red  
 MENU: 4,32,41 Blue on Gray Lt\_Red  
 MENU: 6,32,41 Blue on Gray Lt\_Red

1 Blue  
 1 Blue  
 1 Blue

AND DO: Test Expression  
 data=1  
 + AND [ 3] display array  
 AND DO: Restart Rule  
 OR DO: Test Expression  
 data=2  
 + AND [ 2] display allarray  
 AND DO: Restart Rule  
 OR DO: Test Expression  
 data=3  
 + AND [ 5] display number  
 AND DO: Restart Rule  
 OR DO: Test Expression  
 data=4  
 AND DO: Succeed

[ 3] display array  
 IF

DO: Menu Question menu\$  
 Display Data

\*  
\* RULE LIST  
\*

	NEXT	{		}	
	PREVIOUS	{		}	
	CONTINUE	{		}	
	COL : SURR		White on Blue		
	COL : 0,0		White on Blue		
1 Blue	MENU: 2,18,28		Blue on Gray		Lt_Red
1 Blue	MENU: 4,18,28		Blue on Gray		Lt_Red
1 Blue	MENU: 6,18,28		Blue on Gray		Lt_Red

AND DO: Test Expression  
       start("N",menu\$)  
 + AND [ 18] next line  
   AND DO: Restart Rule  
  
 OR DO: Test Expression  
       start("P",menu\$)  
 + AND [ 20] previous line  
   AND DO: Restart Rule

[ 18] next line		Sp
IF	DO: Test Expression	
	number<(totalnumber-1)	
AND	DO: Assign Variable	
	number:=number+1	
OR	DO: Assign Variable	
	number:=0	

[ 20] previous line		Sp
IF	DO: Test Expression	
	number>0	
AND	DO: Assign Variable	
	number:=number-1	
OR	DO: Assign Variable	
	number:=(totalnumber-1)	

[ 2] display allarray		Sp
IF	DO: Menu Question menu\$	
	Would you like to see :	

	Next page of the array	{		}	
	Previous page of the array	{		}	
	COL : SURR		White on Blue		
	COL : 0,0		White on Blue		
Blue	MENU: 3,35,45		Blue on Gray		Lt_Red
Blue	MENU: 5,35,45		Blue on Gray		Lt_Red

```

*
*
* -----
      AND      DO: Test Expression
                start("N",menu$)
+   AND [ 19] next page
      AND      DO: Restart Rule

      OR      DO: Test Expression
                start("P",menu$)
+   AND [ 21] previous page
      AND      DO: Restart Rule

[ 19] next page      Sp
      IF      DO: Test Expression
                index<(size-displayed)
      AND      DO: Assign Variable
                index:=index+displayed

      OR      DO: Succeed

[ 21] previous page      Sp
      IF      DO: Test Expression
                index>0
      AND      DO: Assign Variable
                index:=index-displayed

      OR      DO: Succeed

[ 5] display number      Sp
      IF      DO: Assign Variable
                number:=number+1
      AND      DO: Assign Variable
                old_value:=obs[number]
      AND      DO: Assign Variable
                text$:=text$+" "+string$(obs[number],9,4)
      AND      DO: Display Form
                This array is      [name$      ]
                OLD VALUE WAS      [old_valu]
                NEW VALUE IS      [new_valu]
                NUMBER IS      [number ]
                TOTAL NUMBER IS      [totalnum]
                Press ( ENTER) to display next value
COL : SURR      White on Blue
COL : 0,0      White on Blue
OUT : 0,24,49
      name$
OUT : 2,24,33,0
      old_value
OUT : 4,24,33,0
      new_value

```

\*  
\* RULE LIST  
\* -----

Mon Aug 14 06:46:43 1995 Page: 8

OUT : 6,24,33,0  
number  
OUT : 8,24,33,0  
totalnumber

AND DO: Display Form  
[text\$ ]  
COL : SURR White on Blue  
COL : 0,0 White on Blue  
OUT : 0,0,49  
text\$

AND DO: Test Expression  
number<totalnumber

AND DO: Restart Rule

OR DO: Succeed

\* RULE LIST

Mon Aug 14 06:42:23 1995 Page: 1

[ 12]

IF

CRYSTAL MASTER RULE  
DO: Menu Question zIR  
WHAT AREA

DATA { }

MODEL { }

OBJECTIVE { }

FINISH { }

COL : SURR White on Blue

COL : 0,0 White on Blue

MENU: 2,10,13 Blue on Gray

Lt\_Red

MENU: 4,10,13 Blue on Gray

Lt\_Red

MENU: 6,10,13 Blue on Gray

Lt\_Red

MENU: 8,10,13 Blue on Gray

Lt\_Red

AND DO: Test Expression

zIR=1

AND DO: Assign Variable

NFI:=1

+ AND [ 4] DATA

AND DO: Restart Rule

OR DO: Test Expression

zIR=2

+ AND [ 7] MODEL

AND DO: Display Form

MODEL IS [MODEL\$

COL : SURR White on Blue

COL : 0,0 White on Blue

OUT : 0,9,38,0

MODEL\$

AND DO: Restart Rule

OR DO: Test Expression

zIR=3

+ AND [ 3] CURRENT STATE

+ AND [ 8] OBJECTIVE

AND DO: Restart Rule

+ OR [ 5] FINISH

OR DO: Menu Question zIR

WHAT AREA

DATA { }

MODEL { }

OBJECTIVE { }

FINISH { }

COL : SURR White on Blue

COL : 0,0 White on Blue

MENU: 2,10,13 Blue on Gray

Lt\_Red

Blue

\*  
\* RULE LIST  
\*

Mon Aug 14 06:42:24 1995 Page: 2

n Blue	MENU: 4,10,13	Blue on Gray	Lt_Red
n Blue	MENU: 6,10,13	Blue on Gray	Lt_Red
n Blue	MENU: 8,10,13	Blue on Gray	Lt_Red

```

      AND      DO: Test Expression
                zIR=1
      AND      DO: Assign Variable
                NFI:=1
+   AND [ 4] DATA
      AND      DO: Restart Rule

      OR      DO: Test Expression
                zIR=2
+   AND [ 7] MODEL
      AND      DO: Display Form
                MODEL IS [MODEL$
                COL : SURR      White on Blue
                COL : 0,0      White on Blue
                OUT : 0,9,38,0
                MODEL$
      AND      DO: Restart Rule

      OR      DO: Test Expression
                zIR=3
+   AND [ 3] CURRENT STATE
+   AND [ 8] OBJECTIVE
      AND      DO: Restart Rule

+ OR [ 5] FINISH

```

```

[ 4] DATA
+ IF [ 3] CURRENT STATE
+ AND [ 1] CHRONOLOGICAL ORDERED
+ AND [ 6] IDENTICAL BUT NOT NEC INDEP
+ AND [ 10] RENEWAL PROCESS
+ AND [ 2] CONSTANT HAZARD

+ OR [ 9] OTHER DISTRIBUTIONS

+ OR [ 3] CURRENT STATE
+ AND [ 1] CHRONOLOGICAL ORDERED
+ AND [ 6] IDENTICAL BUT NOT NEC INDEP
+ AND [ 10] RENEWAL PROCESS
+ AND [ 2] CONSTANT HAZARD

+ OR [ 9] OTHER DISTRIBUTIONS

```

```

[ 3] CURRENT STATE
      IF
      DO: Display Form
        CURRENT KNOWLEDGE
        DATA IS CHRONOLOGICALLY ORDERED
        DATA HAS TREND
      STATE
        [DCO$]
        [TREND$]

```

\*  
\* RULE LIST  
\* -----

Mon Aug 14 06:42:25 1995 Page: 3

DATA IS DEPENDENT	[DEP\$]
DATA HAS CONSTANT HAZARD	[HAZ\$]
MODEL IS	[MODEL\$]
COL : SURR	White on Blue
COL : 0,0	White on Blue
OUT : 2,42,47,0	
DCO\$	
OUT : 3,42,49,0	
TREND\$	
OUT : 4,42,47,0	
DEP\$	
OUT : 5,42,47,0	
HAZ\$	
OUT : 7,35,49,0	
MODEL\$	

OR

DO: Display Form	STATE
CURRENT KNOWLEDGE	
DATA IS CHRONOLOGICALLY ORDERED	[DCO\$]
DATA HAS TREND	[TREND\$]
DATA IS DEPENDENT	[DEP\$]
DATA HAS CONSTANT HAZARD	[HAZ\$]
MODEL IS	[MODEL\$]
COL : SURR	White on Blue
COL : 0,0	White on Blue
OUT : 2,42,47,0	
DCO\$	
OUT : 3,42,49,0	
TREND\$	
OUT : 4,42,47,0	
DEP\$	
OUT : 5,42,47,0	
HAZ\$	
OUT : 7,35,49,0	
MODEL\$	

[ 1] CHRONOLOGICAL ORDERED	Sp
IF	DO: Test Expression
	NFI=1
AND	DO: Test Expression
	DCO\$="YES"
OR	DO: Yes/No Question
	NO FURTHER INFORMATION
	COL : SURR White on Blue
	COL : 0,0 White on Blue
AND	DO: Assign Variable
	NFI:=1
OR	DO: Assign Variable



```

AND      NFI:=0
          DO: Test Expression
          DCO$="YES"

OR        DO: Yes/No Question
          IS THE DATA CHRONOLOGICALLY ORDERED?
          COL : SURR      White on Blue
          COL : 0,0      White on Blue

AND      DO: Assign Variable
          DCO$="YES"
AND      DO: Conclusion Display

OR        DO: Assign Variable
          DCO$="NO"

OR        DO: Test Expression
          NFI=1
AND      DO: Test Expression
          DCO$="YES"

OR        DO: Yes/No Question
          NO FURTHER INFORMATION
          COL : SURR      White on Blue
          COL : 0,0      White on Blue

AND      DO: Assign Variable
          NFI=1

OR        DO: Assign Variable
          NFI=0
AND      DO: Test Expression
          DCO$="YES"

OR        DO: Yes/No Question
          IS THE DATA CHRONOLOGICALLY ORDERED?
          COL : SURR      White on Blue
          COL : 0,0      White on Blue

AND      DO: Assign Variable
          DCO$="YES"
AND      DO: Conclusion Display

OR        DO: Assign Variable
          DCO$="NO"

```

```

[ 6] IDENTICAL BUT NOT NEC INDEP      Sp
    IF      DO: Test Expression
            NFI=1

OR        DO: Yes/No Question
          NO FURTHER INFORMATION
          COL : SURR      White on Blue
          COL : 0,0      White on Blue

AND      DO: Assign Variable

```

NFI:=1

```

OR      DO: Display Form
        TEST THE TREND USING EITHER LAPLACE OR
        MIL-HDBK-219
        COL : SURR      White on Blue
        COL : 0,0       White on Blue

AND     DO: Yes/No Question
        IS THERE A TREND
        COL : SURR      White on Blue
        COL : 0,0       White on Blue

AND     DO: Assign Variable
        TREND$:="YES"

AND     DO: Conclusion Display

OR      DO: Assign Variable
        TREND$:="NO"

OR      DO: Test Expression
        NFI=1

OR      DO: Yes/No Question
        NO FURTHER INFORMATION
        COL : SURR      White on Blue
        COL : 0,0       White on Blue

AND     DO: Assign Variable
        NFI:=1

OR      DO: Display Form
        TEST THE TREND USING EITHER LAPLACE OR
        MIL-HDBK-219
        COL : SURR      White on Blue
        COL : 0,0       White on Blue

AND     DO: Yes/No Question
        IS THERE A TREND
        COL : SURR      White on Blue
        COL : 0,0       White on Blue

AND     DO: Assign Variable
        TREND$:="YES"

AND     DO: Conclusion Display

OR      DO: Assign Variable
        TREND$:="NO"

```

[ 10] RENEWAL PROCESS

Sp

```

IF      DO: Test Expression
        NFI=1

OR      DO: Yes/No Question
        NO FURTHER INFORMATION
        COL : SURR      White on Blue

```

\*  
\* RULE LIST  
\* -----

Mon Aug 14 06:42:27 1995 Page: 6

	COL : 0,0	White on Blue	
	AND	DO: Assign Variable NFI:=1	
	OR	DO: Display Form TEST FOR DEPENDENCY COL : SURR White on Blue COL : 0,0 White on Blue	
	AND	DO: Menu Question DEP DATA IS INDEPENDENT { } DEPENDENT { } COL : SURR White on Blue COL : 0,0 White on Blue MENU: 0,23,25 Blue on Gray	Lt_Red
n Blue		MENU: 1,23,25 Blue on Gray	Lt_Red
1 Blue			
	AND	DO: Test Expression DEP=1	
	AND	DO: Assign Variable DEP\$:="YES"	
	AND	DO: Conclusion Display	
	OR	DO: Assign Variable DEP\$:="NO"	
	OR	DO: Test Expression NFI=1	
	OR	DO: Yes/No Question NO FURTHER INFORMATION COL : SURR White on Blue COL : 0,0 White on Blue	
	AND	DO: Assign Variable NFI:=1	
	OR	DO: Display Form TEST FOR DEPENDENCY COL : SURR White on Blue COL : 0,0 White on Blue	
	AND	DO: Menu Question DEP DATA IS INDEPENDENT { } DEPENDENT { } COL : SURR White on Blue COL : 0,0 White on Blue MENU: 0,23,25 Blue on Gray	Lt_Red
1 Blue		MENU: 1,23,25 Blue on Gray	Lt_Red
1 Blue			
	AND	DO: Test Expression DEP=1	
	AND	DO: Assign Variable DEP\$:="YES"	
	AND	DO: Conclusion Display	
	OR	DO: Assign Variable	

DEP\$:="NO"

[ 2] CONSTANT HAZARD

Sp

```

IF      DO: Test Expression
        NFI=1

OR      DO: Yes/No Question
        NO FURTHER INFORMATION
        COL : SURR      White on Blue
        COL : 0,0      White on Blue

AND     DO: Assign Variable
        NFI:=1

OR      DO: Yes/No Question
        IS THE HAZARD CONSTANT?
        COL : SURR      White on Blue
        COL : 0,0      White on Blue

AND     DO: Assign Variable
        HAZ$:="YES"

OR      DO: Assign Variable
        HAZ$:="NO"

OR      DO: Test Expression
        NFI=1

OR      DO: Yes/No Question
        NO FURTHER INFORMATION
        COL : SURR      White on Blue
        COL : 0,0      White on Blue

AND     DO: Assign Variable
        NFI:=1

OR      DO: Yes/No Question
        IS THE HAZARD CONSTANT?
        COL : SURR      White on Blue
        COL : 0,0      White on Blue

AND     DO: Assign Variable
        HAZ$:="YES"

OR      DO: Assign Variable
        HAZ$:="NO"

```

[ 9] OTHER DISTRIBUTIONS

```

IF      DO: Display Form
        FIT OTHER DISTRIBUTIONS OR USE
        DISTRIBUTION - FREE TECHNIQUES
        COL : SURR      White on Blue
        COL : 0,0      White on Blue

```

\*  
\* RULE LIST  
\*

Mon Aug 14 06:42:29 1995 Page: 8

```

OR          DO: Display Form
              FIT OTHER DISTRIBUTIONS OR USE
              DISTRIBUTION - FREE TECHNIQUES
COL : SURR      White on Blue
COL : 0,0       White on Blue

```

```

[ 7] MODEL                                     Sp
+ IF [ 3] CURRENT STATE
    AND DO: Test Expression
        DCO$="NO"
    AND DO: Assign Variable
        MODEL$="UNKNOWN"

OR      DO: Test Expression
        TREND$="YES"
    AND DO: Assign Variable
        MODEL$="NHPP"

OR      DO: Test Expression
        DEP$="YES"
    AND DO: Assign Variable
        MODEL$="BPP"

OR      DO: Test Expression
        HAZ$="YES"
    AND DO: Assign Variable
        MODEL$="HPP"

OR      DO: Assign Variable
        MODEL$="OTHER MODEL"

+ OR [ 3] CURRENT STATE
    AND DO: Test Expression
        DCO$="NO"
    AND DO: Assign Variable
        MODEL$="UNKNOWN"

OR      DO: Test Expression
        TREND$="YES"
    AND DO: Assign Variable
        MODEL$="NHPP"

OR      DO: Test Expression
        DEP$="YES"
    AND DO: Assign Variable
        MODEL$="BPP"

OR      DO: Test Expression
        HAZ$="YES"
    AND DO: Assign Variable
        MODEL$="HPP"

OR      DO: Assign Variable
        MODEL$="OTHER MODEL"

```

[ 8] OBJECTIVE  
IF

DO: Menu Question OBJ  
WHAT IS THE AIM OF THIS ENQUIRY ?  
IS IT TO DETERMINE :

TYPE OF DATA MODEL { }

TYPE OF DATA DISTRIBUTION { }

APPROPRIATE DATA TEST { }

COL : SURR White on Blue  
COL : 0,0 White on Blue  
MENU: 4,34,39 Blue on Gray

Lt\_Red

MENU: 6,34,39 Blue on Gray

Lt\_Red

MENU: 8,34,39 Blue on Gray

Lt\_Red

AND DO: Test Expression  
OBJ=1

AND DO: Assign Variable  
NFI:=1

+ AND [ 7] MODEL  
AND DO: Conclusion Display

OR DO: Test Expression  
OBJ=2

+ AND [ 4] DATA  
AND DO: Conclusion Display

OR DO: Test Expression  
OBJ=3

+ AND [ 11] TEST  
AND DO: Conclusion Display  
AND DO: Restart Rule

OR DO: Menu Question OBJ  
WHAT IS THE AIM OF THIS ENQUIRY ?  
IS IT TO DETERMINE :

TYPE OF DATA MODEL { }

TYPE OF DATA DISTRIBUTION { }

APPROPRIATE DATA TEST { }

COL : SURR White on Blue  
COL : 0,0 White on Blue  
MENU: 4,34,39 Blue on Gray

Lt\_Red

MENU: 6,34,39 Blue on Gray

Lt\_Red

MENU: 8,34,39 Blue on Gray

Lt\_Red

AND DO: Test Expression  
OBJ=1

AND DO: Assign Variable  
NFI:=1

```

+   AND [ 7] MODEL
    AND   DO: Conclusion Display

OR      DO: Test Expression
        OBJ=2

+   AND [ 4] DATA
    AND   DO: Conclusion Display

OR      DO: Test Expression
        OBJ=3

+   AND [ 11] TEST
    AND   DO: Conclusion Display
    AND   DO: Restart Rule
  
```

[ 11] TEST  
IF

DO: Menu Question TST  
WHICH ONE OF THOSE MODELS IS THE DATA MODEL :

HPP { }

NHPP { }

BPP { }

COL : SURR White on Blue

COL : 0,0 White on Blue

MENU: 2,15,19 Blue on Gray

Lt\_Red

MENU: 4,15,19 Blue on Gray

Lt\_Red

MENU: 6,15,19 Blue on Gray

Lt\_Red

AND DO: Test Expression

TST=1

AND DO: Display Form

The test which is needed to  
be used to decide that the  
data model is an HPP model  
is a CONSTANT HAZARD TEST.

COL : SURR White on Blue

COL : 0,0 White on Blue

AND DO: Conclusion Display

OR DO: Test Expression

TST=2

AND DO: Display Form

The test which is needed to  
be used to decide that the  
data model is an NHPP test  
is a TREND TEST.

COL : SURR White on Blue

COL : 0,0 White on Blue

AND DO: Conclusion Display

OR DO: Test Expression

TST=3

AND DO: Display Form

1 Blue

1 Blue

1 Blue



\*  
\* RULE LIST  
\* -----

Mon Aug 14 06:42:31 1995 Page: 11

The test which is needed to  
be used to decide that the  
data model is an BPP model  
is a DEPENDENCE TEST.

COL : SURR White on Blue  
COL : 0,0 White on Blue

AND DO: Restart Rule

OR DO: Menu Question TST  
WHICH ONE OF THOSE MODELS IS THE DATA MODEL :

HPP { }

NHPP { }

BPP { }

COL : SURR White on Blue  
COL : 0,0 White on Blue  
MENU: 2,15,19 Blue on Gray

Lt\_Red

MENU: 4,15,19 Blue on Gray

Lt\_Red

MENU: 6,15,19 Blue on Gray

Lt\_Red

1 Blue

1 Blue

1 Blue

AND DO: Test Expression  
TST=1

AND DO: Display Form  
The test which is needed to  
be used to decide that the  
data model is an HPP model  
is a CONSTANT HAZARD TEST.

COL : SURR White on Blue  
COL : 0,0 White on Blue

AND DO: Conclusion Display

OR DO: Test Expression  
TST=2

AND DO: Display Form  
The test which is needed to  
be used to decide that the  
data model is an NHPP test  
is a TREND TEST.

COL : SURR White on Blue  
COL : 0,0 White on Blue

AND DO: Conclusion Display

OR DO: Test Expression  
TST=3

AND DO: Display Form  
The test which is needed to  
be used to decide that the  
data model is an BPP model  
is a DEPENDENCE TEST.

COL : SURR White on Blue  
COL : 0,0 White on Blue

AND DO: Restart Rule

\*  
\* RULE LIST  
\* -----

Mon Aug 14 06:42:32 1995 Page: 12

[ 5] FINISH  
IF  
OR

DO: Conclusion Display

DO: Conclusion Display

[ 23]

IF

CRYSTAL MASTER RULE  
DO: Menu Question gb

Is the general distribution

Blue

Blue

GB1 {	}	or	GB2 {	}
COL : SURR			White on Blue	
COL : 0,0			White on Blue	
MENU: 5,9,16			Blue on Gray	Lt_Red
MENU: 5,34,41			Blue on Gray	Lt_Red

AND DO: Test Expression  
gb=1

+ AND [ 9] GB1

OR DO: Test Expression  
gb=2

+ AND [ 10] GB2

OR DO: Display Form

More information is required

COL : SURR	White on Blue
COL : 0,0	White on Blue

[ 9] GB1

IF

DO: Menu Question b1

Does a =1 and the Inverse form of the  
distribution is obtained if the sign of  
is changed ?

Blue

Blue

YES {	}	NO {	}
COL : SURR		White on Blue	
COL : 0,0		White on Blue	
MENU: 8,9,17		Blue on Gray	Lt_Red
MENU: 8,33,42		Blue on Gray	Lt_Red

AND DO: Test Expression  
b1=1

+ AND [ 1] B1

OR DO: Test Expression  
b1=2

AND DO: Menu Question gg

\*  
\* RULE LIST  
\* -----

Mon Aug 14 06:34:10 1995 Page: 2

1/a  
Is q infinitive with b=q B

	Yes { }	NO { }	
	COL : SURR	White on Blue	
	COL : 0,0	White on Blue	
1 Blue	MENU: 7,9,15	Blue on Gray	Lt_Red
1 Blue	MENU: 7,32,38	Blue on Gray	Lt_Red

AND DO: Test Expression  
gg=1  
+ AND [ 11] GG  
OR DO: Display Form

The distribution is GB1  
COL : SURR White on Blue  
COL : 0,0 White on Blue

[ 1] B1  
IF DO: Menu Question b1

Does q=1

	YES { }	NO { }	
	COL : SURR	White on Blue	
	COL : 0,0	White on Blue	
1 Blue	MENU: 7,9,17	Blue on Gray	Lt_Red
1 Blue	MENU: 7,31,40	Blue on Gray	Lt_Red

AND DO: Test Expression  
b1=1  
+ AND [ 17] P  
OR DO: Test Expression  
b1=2  
AND DO: Menu Question ga

1/a  
Is q infinitive with b=q B

YES { }	NO { }
COL : SURR	White on Blue

\* RULE LIST  
\* -----

Mon Aug 14 06:34:11 1995 Page: 3

	COL : 0,0	White on Blue	
1 Blue	MENU: 8,8,14	Blue on Gray	Lt_Red
1 Blue	MENU: 8,30,38	Blue on Gray	Lt_Red

AND DO: Test Expression  
ga=1  
+ AND [ 8] GA  
OR DO: Display Form

The distribution is B1  
COL : SURR White on Blue  
COL : 0,0 White on Blue

[ 17] P IF DO: Menu Question u

Does p=1

	YES {		NO {	
	COL : SURR	White on Blue		
	COL : 0,0	White on Blue		
1 Blue	MENU: 8,11,19	Blue on Gray		Lt_Red
1 Blue	MENU: 8,32,42	Blue on Gray		Lt_Red

AND DO: Test Expression  
u=1  
+ AND [ 20] U  
OR DO: Display Form

The distribution is P  
COL : SURR White on Blue  
COL : 0,0 White on Blue

[ 20] U IF DO: Display Form

\*  
\* RULE LIST  
\* -----

Mon Aug 14 06:34:11 1995 Page: 4

The distgribution is U  
COL : SURR White on Blue  
COL : 0,0 White on Blue

[ 8] GA  
IF DO: Menu Question ep

Does p=1

	YES {	}	or	NO {	}	
	COL : SURR			White on Blue		
	COL : 0,0			White on Blue		
1 Blue	MENU: 7,8,16			Blue on Gray		Lt_Red
1 Blue	MENU: 7,32,39			Blue on Gray		Lt_Red

AND DO: Test Expression  
ep=1  
+ AND [ 5] EXP  
OR DO: Test Expression  
ep=1  
AND DO: Menu Question x2

Does B = 2

	YES {	}	or	NO {	}	
	COL : SURR			White on Blue		
	COL : 0,0			White on Blue		
1 Blue	MENU: 7,8,16			Blue on Gray		Lt_Red
1 Blue	MENU: 7,32,40			Blue on Gray		Lt_Red

AND DO: Test Expression  
x2=1  
+ AND [ 22] X2  
OR DO: Display Form

The distribution is GA  
COL : SURR White on Blue  
COL : 0,0 White on Blue

[ 5] EXP  
IF DO: Display Form

The distribution is EXP  
COL : SURR White on Blue  
COL : 0,0 White on Blue

[ 22] X2  
IF DO: Display Form

The distribution is 2  
COL : SURR X White on Blue  
COL : 0,0 White on Blue

[ 11] GG  
IF DO: Menu Question n

Does a=2 and p=1/2

	YES {	or	NO {	
	COL : SURR		White on Blue	
	COL : 0,0		White on Blue	
Blue	MENU: 7,7,14		Blue on Gray	Lt_Red
Blue	MENU: 7,31,39		Blue on Gray	Lt_Red

AND DO: Test Expression  
n=1  
+ AND [ 16] N(0,02)  
OR DO: Test Expression  
n=2  
AND DO: Menu Question w

Does P=1 and the Inverse form is obtained if the sign of a is changed ?

	YES { }	NO { }	
	COL : SURR	White on Blue	
	COL : 0,0	White on Blue	
1 Blue	MENU: 7,9,15	Blue on Gray	Lt_Red
1 Blue	MENU: 7,30,36	Blue on Gray	Lt_Red

AND DO: Test Expression  
w=1  
+ AND [ 21] W  
OR DO: Test Expression  
w=2  
AND DO: Menu Question nl

Is a  $\sim 0$  with  $b = (o a)^{2 2 1/a}$ ,  $p = (au=1)/o a^{2 2}$

	YES { }	NO { }	
	COL : SURR	White on Blue	
	COL : 0,0	White on Blue	
Blue	MENU: 7,10,16	Blue on Gray	Lt_Red
Blue	MENU: 7,28,34	Blue on Gray	Lt_Red

AND DO: Test Expression  
nl=1  
+ AND [ 14] LN  
OR DO: Test Expression  
nl=2  
AND DO: Menu Question ga

Does  $a=1$  and the inverse form is obtained if the sign of  $a$  is changed?

	YES { }	NO { }	
	COL : SURR	White on Blue	
	COL : 0,0	White on Blue	
Blue	MENU: 7,10,16	Blue on Gray	Lt_Red
Blue	MENU: 7,31,38	Blue on Gray	Lt_Red

AND DO: Test Expression  
ga=1  
+ AND [ 8] GA  
OR DO: Display Form

The distribution is GG  
COL : SURR White on Blue  
COL : 0.0 White on Blue



[ 16] N(0,0<sup>2</sup>)  
IF

DO: Display Form

The distribution is N(0,0<sup>2</sup>)  
COL : SURR White on Blue  
COL : 0,0 White on Blue

[ 21] W  
IF

DO: Menu Question w3

Does a=1 and the inverse form is obtained if the  
sign of a is changed ?

YES {	or	NO {	
COL : SURR	White on Blue		
COL : 0,0	White on Blue		
MENU: 7,8,15	Blue on Gray		Lt_Red
MENU: 7,33,40	Blue on Gray		Lt_Red

Blue

Blue

AND DO: Test Expression  
w3=1  
+ AND [ 5] EXP  
OR DO: Test Expression  
w3=2  
AND DO: Menu Question r

Does a=2 and the Inverse form is obtained if  
the sign of a is changed ?

YES {	or	NO {	
COL : SURR	White on Blue		
COL : 0,0	White on Blue		
MENU: 8,8,15	Blue on Gray		Lt_Red
MENU: 8,34,40	Blue on Gray		Lt_Red

Blue

Blue

AND DO: Test Expression  
r=1  
+ AND [ 18] R  
OR DO: Test Expression

AND r=2  
DO: Display Form

The distribution is W.  
COL : SURR White on Blue  
COL : 0,0 White on Blue

[ 18] R IF DO: Display Form

The distribution is R  
COL : SURR White on Blue  
COL : 0,0 White on Blue

[ 14] LN IF DO: Display Form

Then distribution is LN  
COL : SURR White on Blue  
COL : 0,0 White on Blue

[ 10] GB2 IF

DO: Menu Question gb2  
Choose the correct model:  
{ }1- q=1  
{ }2- a=2, b=v, p=1/2, q=v/2  
{ }3- p=1  
{ }4- a=1 1/a  
{ }5- q<sup>infinity</sup> with b=q<sup>2</sup> B and the inverse  
form of the distribution is obtained  
2 2 1/a 2 2  
{ }6- a<sup>0</sup> with b=(qo a), p=(au=1)/o a  
{ }7- None of the above  
COL : SURR White on Blue  
COL : 0,0 White on Blue  
MENU: 1,0,7 Blue on Gray Lt\_Red  
MENU: 2,0,7 Blue on Gray Lt\_Red  
MENU: 3,0,7 Blue on Gray Lt\_Red  
MENU: 4,0,7 Blue on Gray Lt\_Red  
MENU: 5,0,7 Blue on Gray Lt\_Red

Blue  
Blue  
Blue  
Blue  
Blue

\*  
\* RULE LIST  
\*

Mon Aug 14 06:34:15 1995 Page: 9

1 Blue	MENU: 8,0,7	Blue on Gray	Lt_Red
1 Blue	MENU: 9,0,7	Blue on Gray	Lt_Red

```

      AND      DO: Test Expression
                gb2=1
+   AND [  4] BR3
      OR      DO: Test Expression
                gb2=2
+   AND [ 19] t
      OR      DO: Test Expression
                gb2=3
+   AND [  3] BR12
      OR      DO: Test Expression
                gb2=4
+   AND [  2] B2
      OR      DO: Test Expression
                gb2=5
+   AND [ 11] GG
      OR      DO: Test Expression
                gb2=6
+   AND [ 15] LT
      OR      DO: Test Expression
                gb2=7
      AND      DO: Display Form
  
```

The distribution is GB2  
COL : SURR            White on Blue  
COL : 0,0            White on Blue

[ 4] BR3  
IF                    DO: Menu Question   fis

Does p=1

	YES    {		}   or   NO   {		}
	COL : SURR		White on Blue		
	COL : 0,0		White on Blue		
1 Blue	MENU: 6,9,17		Blue on Gray		Lt_Red
1 Blue	MENU: 6,31,40		Blue on Gray		Lt_Red

```

      AND      DO: Test Expression
                fis=1
+   AND [  7] FISK
  
```

\*  
\* RULE LIST  
\* -----

Mon Aug 14 06:34:16 1995 Page: 10

OR DO: Test Expression  
fis=2  
AND DO: Menu Question 12

Does a=-1 ?

	YES {	}	or	NO {	}
	COL : SURR			White on Blue	
	COL : 0,0			White on Blue	
1 Blue	MENU: 7,9,16			Blue on Gray	Lt_Red
1 Blue	MENU: 7,30,38			Blue on Gray	Lt_Red

AND DO: Test Expression  
12=1  
+ AND [ 12] L  
OR DO: Display Form

The distribution is BR3.  
COL : SURR White on Blue  
COL : 0,0 White on Blue

[ 7] FISK  
IF DO: Display Form

The distribution is FISK.  
COL : SURR White on Blue  
COL : 0,0 White on Blue

[ 12] L  
IF DO: Menu Question ep2

Is  $q \sim \frac{1}{a}$  infinity with  $b=q$  B ?

	YES {	}	or	NO {	}
	COL : SURR			White on Blue	
	COL : 0,0			White on Blue	
1 Blue	MENU: 6,9,17			Blue on Gray	Lt_Red

RULE LIST

Mon Aug 14 06:34:17 1995 Page: 11

Blue MENU: 6,31,39 Blue on Gray Lt\_Red

AND DO: Test Expression  
ep2=1  
+ AND [ 5] EXP  
OR DO: Display Form

The distribution is L.  
COL : SURR White on Blue  
COL : 0,0 White on Blue

[ 19] t IF DO: Display Form

The distribution is t.  
COL : SURR White on Blue  
COL : 0,0 White on Blue

[ 3] BR12 IF DO: Menu Question 1

Does a=1 ?

YES { } or NO { }  
COL : SURR White on Blue  
COL : 0,0 White on Blue  
MENU: 6,9,15 Blue on Gray Lt\_Red  
MENU: 6,31,38 Blue on Gray Lt\_Red

AND DO: Test Expression  
l=1  
+ AND [ 12] L  
OR DO: Test Expression  
l=2  
AND DO: Menu Question 11

does q=1 ?

		Yes {		} or NO {		}
	COL :	SURR		White on Blue		
	COL :	0,0		White on Blue		
1 Blue	MENU:	7,10,18		Blue on Gray		Lt_Red
1 Blue	MENU:	7,30,38		Blue on Gray		Lt_Red

AND DO: Test Expression  
11=1  
+ AND [ 7] FISK  
OR DO: Test Expression  
11=2  
AND DO: Menu Question w1

1/a  
Does q~infinity with b=p B, the inverse form  
the distribution is obtained.

		YES {		} or No {		}
	COL :	SURR		White on Blue		
	COL :	0,0		White on Blue		
1 Blue	MENU:	8,8,16		Blue on Gray		Lt_Red
1 Blue	MENU:	8,30,38		Blue on Gray		Lt_Red

AND DO: Test Expression  
w1=1  
+ AND [ 21] W  
OR DO: Test Expression  
w1=2  
AND DO: Display Form

The distribution is BR12.  
COL : SURR White on Blue  
COL : 0,0 White on Blue

[ 2] B2 IF DO: Menu Question 13

Does p=1 ?

YES { } or NO { }

RULE LIST

Mon. Aug 14 06:34:18 1995 Page: 13

	COL : SURR	White on Blue	
	COL : 0,0	White on Blue	
1 Blue	MENU: 7,6,10	Blue on Gray	Lt_Red
1 Blue	MENU: 7,28,33	Blue on Gray	Lt_Red

AND DO: Test Expression  
13=1  
+ AND [ 12] L  
OR DO: Test Expression  
13=2  
AND DO: Menu Question f

Does  $b=v/u$  ,  $p=u/2$  and  $q=v/2$  ?

	YES {	}	or	NO {	}	
	COL : SURR	White on Blue				
	COL : 0,0	White on Blue				
Blue	MENU: 7,8,15	Blue on Gray				Lt_Red
Blue	MENU: 7,29,35	Blue on Gray				Lt_Red

AND DO: Test Expression  
f=1  
+ AND [ 6] F  
OR DO: Test Expression  
f=2  
AND DO: Menu Question b2

Is  $q \sim \infty$  with  $b=q^{1/a}$  B

	YES {	}	or	NO	
	COL : SURR	White on Blue			
	COL : 0,0	White on Blue			
Blue	MENU: 7,9,15	Blue on Gray			Lt_Red

AND DO: Test Expression  
b2=1  
+ AND [ 8] GA  
OR DO: Display Form

The disribution is B2.  
COL : SURR White on Blue  
COL : 0,0 White on Blue

\*  
\* RULE LIST  
\*

Mon Aug 14 06:34:19 1995 Page: 14

[ 6] F IF

DO: Menu Question fl

Is q~infinity ?

YES { } or NO { }

COL : SURR White on Blue

COL : 0,0 White on Blue

MENU: 7,8,14 Blue on Gray

Lt\_Red

1 Blue

MENU: 7,29,35 Blue on Gray

Lt\_Red

1 Blue

AND DO: Test Expression

fl=1

+ AND [ 22] X2

OR DO: Display Form

The distribution is F.

COL : SURR White on Blue

COL : 0,0 White on Blue

[ 15] LT

IF DO: Menu Question lt

Is q~infinity ?

YES { } or NO { }

COL : SURR White on Blue

COL : 0,0 White on Blue

MENU: 7,7,14 Blue on Gray

Lt\_Red

1 Blue

MENU: 7,28,34 Blue on Gray

Lt\_Red

1 Blue

AND DO: Test Expression

lt=1

+ AND [ 14] LN

OR DO: Test Expression

lt=2

AND DO: Menu Question lc



\*  
\* RULE LIST  
\* -----

Mon Aug 14 06:34:20 1995 Page: 15

Does LC=1/2 ?

	YES {	or NO {	
COL : SURR			White on Blue
COL : 0,0			White on Blue
MENU: 8,9,15			Blue on Gray
			Lt_Red
1 Blue			
MENU: 8,31,38			Blue on Gray
			Lt_Red
1 Blue			

AND DO: Test Expression  
lc=1  
+ AND [ 13] LC  
OR DO: Display Form

The distribution is LT.  
COL : SURR White on Blue  
COL : 0,0 White on Blue

[ 13] LC IF DO: Display Form

The distribution is LC.  
COL : SURR White on Blue  
COL : 0,0 White on Blue

## **APPENDIX FIVE**

The following is a listing of all ARDA's functions in APL2.

```

[ 0]ABSTRACT
[ 1]2 1+CR 'ABSTRACT'
[ 2]R (C) Copyright IBM Corp 1984, 1988.
[ 3]R
[ 4]R THE DISPLAY WORKSPACE.
[ 5]R -----
[ 6]R
[ 7]R This workspace contains the DISPLAY function which may
[ 8]R be used to display an APL2 array in an easy-to-understand
[ 9]R pictorial format.
[10]R

[0]ARDA
[1]'WELCOME TO ARDA'
[2]' '
[3]'AN EXPERT SYSTEM FOR RELIABILITY DATA ANALYSIS'
[4]'COPYWRITE JAKE ANSELL AND MULHIM AL-DOORI'
[5]' '
[6]'PLEASE ENTER THE COMMAND START TO BEGIN A SESSION WITH ARDA'

[0]CALLDAT
[1]R THE AIM OF THIS FUNCTION IS TO DISPLAY A MESSAGE TO THE USER
[2]R WHEN DATA WORKSPACE IS USED OR ADDED TO THE CURRENT WORKSPACE
[3]' '
[4]'WELCOME TO THE RDADATA WORKSPACE'
[5]' '
[6]'FOR DATA ANALYSIS ENTER THE COMMAND- DATA '

[0]CALLDIST
[1]R THE AIM OF THIS FUNCTION IS TO DISPLAY A MESSAGE OF INSTRUCTIONS
[2]R TO THE USER WHEN RDADIST WORKSPACE IS USED
[3]' '
[4]'WELCOME TO RDADIST'
[5]' '

```

[6]'IF YOU WANT TO FIT DISTRIBUTION TO DATA ENTER COMMAND- FIT\_DIST\_ALL

[7]' '

[8]'IF YOU WANT TO TEST A PARTICULAR DISTRIBUTION ENTER COMMAND - DIS

[0]CALLMODEL

[1]THE AIM OF THIS FUNCTION IS TO DISPLAY A MESSAGE OF INSTRUCTIONS TO

[2]USER WHEN THIS WORKSPACE IS LOADED

[3]' '

[4]'WELCOME TO RDAMODEL'

[5]' '

[6]'IF YOU WANT TO FIT A MODEL TO DATA ENTER COMMAND MODELS '

[0]CHANGE C

[1]' ENTER VALUE'

[2]A[',(C[1]),',';','(C[2]),'] $\leftarrow$  ,

[0]A $\leftarrow$ CHECK\_LIFETIME X

[1]FN:CHECK INPUT FOR LIFETIME DISTRIBUTION TEST

[2]RA:DATA FOR TEST

[3]LA:NO LEFT ARGUMENT

[0]A $\leftarrow$ CHECK\_LIKELIHOOD X

[1]FN:CHECK THE INPUT FOR LIKELIHOOD TEST

[2]RA:DATA FOR TEST

[3]LA:NO LEFT ARGUMENT

[0]A $\leftarrow$ CHECK\_PDF X

[1]FN:CHECK THE INPUT TO THE PROBABILITY DENSITY FUNCTION TEST

[2]RA:DATA FOR TEST

[3]LA:NO LEFT ARGUMENT

[4]s\_u $\leftarrow$ ((+/(X-s\_u)\*2 $\div$ -1+ $\rho$ X)\*0.5),s\_u $\leftarrow$ +/(X $\div$  $\rho$ X

[5]s\_u NORMPDF X

[ 0]RES $\leftarrow$ DEG CHICDF1 CHI;LOGG;J;P;Y;C;S;N;I

[ 1]FN: calculate cumulative function values of the

[ 2]FN: chi squared distribution

[ 3]SS: iso/bs

[ 4]RA: scalar chi value

[ 5]LA: scalar degree of freedom

```

[ 6]A RE: scalar chi cumulative distribution function value
[ 7]A
[ 8]A □ELX←'□ERROR ''ARGUMENT ERROR''
[ 9]P←0.5×DEG
[10]Y←0.5×CHI
[11]C←S+1
[12]N←0
[13]loop:N←N+1
[14]S←S+C×C×Y÷P+N
[15]→(C≥1E-8)↑loop
[16]A calculate logamma of deg
[17]J←0
[18]I←2|DEG
[19]→(DEG≤2)↑cnt
[20]J←+/(0.5×I)+1L (DEG÷2)-I
[21]cnt:LOGG←J+I×0.57236494
[22]RES←|(P×Y)+(S)-Y+(P)+LOGG
[ 0]RES←DEG CHICDF2 CHI;LOGG;J;P;Y;C;S;N;I;M
[ 1]A FN: calculate cumulative function values of the chi
[ 2]A FN: squared distribution
[ 3]A SS: iso/bs
[ 4]A RA: scalar chi squared value
[ 5]A LA: scalar degree of freedom
[ 6]A RE: scalar chi squared cumulative distribution function value
[ 7]A
[ 8]A □ELX←'□ERROR ''ARGUMENT ERROR''
[ 9]□CT←0
[10]P←0.5×DEG
[11]Y←0.5×,CHI↑1E-30
[12]S←1♦M←10♦C←((P×Y),M)÷1
[13]N←0
[14]loop:
[15]S←S++/C+C[;M/M]××\Y÷P+N+1M
[16]N←N+M

```

```

[17] → ((I/O,C[;M]) ≥ 1E-9) → loop
[18] A calculate loggamma of deg
[19] LOGG ← LOGGAMMA DF DEG
[20] RES ← |*(P×Y)+(S)-Y+(P)+LOGG
[21] RES ← (CHI) RES
[0] RES ← DEG CHICDFM X
[1] RES ← 0
[2] J ← 0
[3] 1: RES ← RES, DEG CHICDF1 X[J+J+1]
[4] → (J < X)/1
[0] RES ← CHICONT TABLE; Z
[1] A FN: calculate chi-squared value of a contingency table
[2] A FN: it is assumed that zero rows and zero columns have been removed
[3] A SS: iso/bs
[4] A RA: matrix representing contingency table
[5] A RE: scalar chi-squared value
[6] A
[7] A ELX ← 'ERROR 'ARGUMENT ERROR''
[8] RES ← +/, ((TABLE-Z)*2) ÷ Z + ((+/TABLE) × +/TABLE) ÷ +/, TABLE
[0] RES ← DF CHIDEN X; Y; Z
[1] A FN: returns values of the chi-squared probability density function
[2] A SS: iso/bs
[3] A RA: vector of chi squared values at which probabilities are required
[4] A LA: scalar integer - degrees of freedom of chi-squared distribution
[5] A RE: vector of probabilities
[6] A
[7] A ELX ← 'ERROR 'ARGUMENT ERROR''
[8] RES ← (X*Z) × (*-0.5×X) ÷ (!Z-1+0.5×DF) ÷ 0.5*0.5×DF
[0] RES ← OBS CHIGOF1 EXP
[1] A FN: calculate goodness of fit ie.chi-squared value, using Likelihood
[2] A FN: Statistics
[3] A SS: iso/bs
[4] A RA: vector representing expected values

```

```

[ 5]A LA: vector representing observed values of same length
[ 6]A LA: of right argument
[ 7]A RE: scalar chi squared value
[ 8]A
[ 9]A DELX←'ERROR ' 'ARGUMENT ERROR' '
[10]RES←2×+/OBS×OBS÷EXP
[ 0]RES←OBS CHIGOF2 EXP

[ 1]A FN: calculate goodness of fit ie.chi-squared value, using Pearson's
[ 2]A FN: Statistics
[ 3]A SS: iso/bs
[ 4]A RA: vector representing expected values
[ 5]A LA: vector representing observed values of same length
[ 6]A LA: of right argument
[ 7]A RE: scalar chi squared value
[ 8]A
[ 9]A DELX←'ERROR ' 'ARGUMENT ERROR' '
[10]RES←+/(OBS-EXP)*2)÷EXP
[0]A←CHILDREN C
[1]LOOP
[2]CHECK C=FUNCTION[5]
[3]NOTED THOSE
[0]A←CHILDREN1 C
[1]A←C[1÷4]
[ 0]RES←K CHIQ1 P
[ 1]A FN: Produces Chi Squared quantiles - fast approx.
[ 2]A RA: A vector of values at which quantiles required
[ 3]A RA: values less than 1.
[ 4]A LA: A vector of degrees of freedom of Chi Squared
[ 5]A LA: if not scalar dimensions same as for Right Hand Argument
[ 6]A RE: A vector of quantiles of chi squared same length as
[ 7]A RE: Right Hand Argument.
[ 8]A AL: Approx. BSC P 85
[ 9]A
[10]RES←K×(1-(2÷9×K)-(NORMQ1 P)×(2÷9×K)*2)*3

```

```

[ 0]RES←K CHIQUANT1 P
[ 1]A FN: Produces Chi Squared quantiles - fast approx.
[ 2]A RA: A vector of values at which quantiles required
[ 3]A RA: values less than 1.
[ 4]A LA: A vector of degrees of freedom of Chi Squared
[ 5]A LA: if not scalar dimensions same as for Right Hand Argument
[ 6]A RE: A vector of quantiles of chi squared same length as
[ 7]A RE: Right Hand Argument.
[ 8]A AL: Approx. BSC P 85
[ 9]A
[10]A DELX←'ERROR 'ARGUMENT ERROR''
[11]RES←K×(1-(2÷9×K)-(NORMQUANT1 P)×(2÷9×K)*÷2)*3
[ 0]RES←K CHIQUANT2 X;X1;X2;RES2;ERR
[ 1]A FN: Produces more accurate but slower values of quantiles
[ 2]A FN: of chi squared than CHIQUANT1.
[ 3]A RA: Scalar probability for which quantile required
[ 4]A LA: Scalar degrees of freedom of Chi Squared
[ 5]A RE:Scalar quantile.
[ 6]A
[ 7]A DELX←'ERROR 'ARGUMENT ERROR''
[ 8]ERR←1E-6
[ 9]RES←K×(1-(2÷9×K)-(NORMQUANT1 X)×(2÷9×K)*÷2)*3
[10]loop:X1←1-K CHITAIL1 RES
[11]→(ERR≥|X1-X)/0
[12]RES2←RES×X÷X1
[13]X2←1-K CHITAIL1 RES2
[14]RES←RES-(RES-RES2)×(X-X1)÷X2-X1♦loop
[ 0]RES←DF CHIQUANT3 P;Z;T
[ 1]A FN: calculate chi squared statistic values or quantiles
[ 2]A SS: iso/bs
[ 3]A RA: scalar or vector of probabilities for which
[ 4]A RA: chi squared statistic required
[ 5]A LA: scalar degree of freedom

```



[ 6]A RE: vector of chi squared quantiles of same length as right argument

[ 7]A AL: ?

[ 8]A

[ 9]A DELX←'ERROR ' 'ARGUMENT ERROR' ''

[10]Z←(P÷(1-P)\*2)\*0.5

[11]T←1+(Z\*.3)+.x1E-6×1432788 189269 1308

[12]Z←Z-((Z\*.012)+.x1E-6×2515517 802853 10328)÷T

[13]RES←DF×(1-(2×9×DF)-(Z÷3)×(2÷DF)\*0.5)\*3

[ 0]RES←K CHITAIL1 X;W;A;MASK1;MASK2;N

[ 1]A FN: calculates the tail probabilities for the chi squared distributi

[ 2]A FN: P(X>right argument)

d [ 3]A RA: any shape - point to the right of which probabilities are calcul

[ 4]A LA: number of degrees of freedom (a scalar - common to all RA values)

[ 5]A RE: An array of tail probs. the same size as right hand argument

[ 6]A AL: algorithm based on AS147

[ 7]A

[ 8]A DELX←'ERROR ' 'ARGUMENT ERROR' ''

[ 9]MASK1←X≤54+1.66×K♦X←X×MASK1♦X←X[ MASK2←X≤0

[10]N←[ /, [ 16+1.37×X

[11]K←K÷2♦W←X÷2

[12]A←\*(K×W)-W+LOGGAMMA DF 2×K+1

[13]RES←MASK1×MASK2[ 1-A×1++/x\W\*.÷K+1N

[14]RES←RES×RES≥1E-8

[ 0]RES←DEG CHITAIL2 CHI;LOGG;J;P;Y;C;S;N;I

[ 1]A FN: calculate upper tail probability of chi squared distribution

[ 2]A SS: iso/bs

uired [ 3]R RA: scalar chi squared value for which upper tail probability is req

[ 4]R LA: scalar degree of freedom

[ 5]R RE: scalar probability

[ 6]R AL: ?

[ 7]R

[ 8]R ELX←'ERROR ' 'ARGUMENT ERROR''

[ 9]P←0.5×DEG

[10]Y←0.5×CHI

[11]C←S+1

[12]N←0

[13]loop:N←N+1

[14]S←S+C×C×Y÷P+N

[15]→(C≥1E-8)↑loop

[16]R calculate logamma of deg

[17]J←0

[18]I←2|DEG

[19]→(DEG≤2)↑cnt

[20]J←+/( $-0.5 \times I$ ) + \L (DEG÷2) -  $\sim I$

[21]cnt:LOGG←J+I×0.57236494

[22]RES←|1-(P×S)+ (S)-Y+(S P)+LOGG

[ 0]CHOOSDIST;W;a;b;c;d;e;f

[ 1]'ENTER THE NAME OF THE DISTRIBUTION YOU CHOSE'

[ 2]→('NOR'=W+3↑)/a

```

[ 3]→ ('EXP'=W)/b
[ 4]→ ('GAM'=W)/c
[ 5]→ ('WIE'=W)/d
[ 6]→ ('POI'=W)/e
[ 7]a:RUNNORMEST
[ 8]→0
[ 9]b:RUNEXPOEST
[10]→0
[11]c:RUNGAMMEST
[12]→0
[13]d:RUNWIEBEST
[14]→0
[15]e:RUNPOISSONEST
[ 0]A←CH_INDEP X
[ 1]R FN: CHECK THE INPUT TO THE INDEPENDENCE TEST
[ 2]R RA: DATA FOR TEST
[ 3]R LA: NO LEFT ARGUMENT
[ 4]R
[ 5]R
[ 6]z←(1÷(ρX-1))x+/X
[ 7]u←(1÷(ρX-1))x+/X+1
[ 8]ca←(1÷(ρX-1))x+/(X-z)*2)
[ 9]cb←(1÷(ρX-1))x+/(X+1-u)*2)
[10]t←(1÷(ρX-1))x+/(X*(X+1))-z*u
[11]rj←c÷(ca*cb)*0.5
[12]in←|rj
[13]in>1.96÷(ρX-1)*0.5
[14]→i
[15]'DATA IS INDEPENDENT'
[16]ΔI←1
[17]i:'DATA IS DEPENDENT'
[18]ΔI←0
[0]A←CH_LPTRD X
[1]R FN: CHECK INPUTS TO TREND TEST IF UNACCEPTABLE REPORTS AND LEAVES

```

[2]RA: DATA FOR TEST, VECTOR NUMERIC VALUES  
 [3]RE: A SCALAR 1 IF OK 0 OTHERWISE  
 [4] $A+x/0 \leq X-0, -1+x$   
 [5] $A+Ax/0 \leq X$   
 [6] $A+A+(1-A) \times^{-99}$   
 [0]A+CH\_MILTRD X  
 [1]RFN: CHECK INPUTS TO TREND TEST IF UNACCEPTABLE REPORTS AND LEAVES  
 [2]RA: DATA FOR TEST, VECTOR NUMERIC VALUES  
 [3]RE: A SCALAR 1 IF OK 0 OTHERWISE  
 [4] $A+x/0 \leq X-0, -1+x$   
 [5] $A+Ax/0 \leq X$   
 [6] $A+A+(1-A) \times^{-99}$   
 [ 0]A+COLLECT X  
 [ 1]RFN: TO COLLECT VALUES USER ON MENU CHOICE  
 [ 2]RA: RANGE OF CHOICES A CHARACTER VECTOR  
 [ 3]RE: A CHARACTER SCALAR  
 [ 4]R  
 [ 5]'ENTER NUMBER OF ONE OF OPTIONS'  
 [ 6]L0:A+1+  
 [ 7]+(A+X)/L1  
 [ 8]'ANSWER NOT ACCEPTABLE, USE ONE OF',X  
 [ 9]+L0  
 [10]L1:'YOUR CHOICE IS ',A,' PLEASE CONFIRM Y/N'  
 [11]+('Y'=1+  
 [12]'PLEASE ENTER VALUE YOU WANT'  
 [13]+L0  
 [0]A+COMPARE\_GOODFIT X  
 [1]RNOT ATTEMPTED YET  
 [0]A+COMPARE\_INDEPENDEN X  
 [1]RNOT ATTEMPTED YET  
 [ 0]A+COMPARE\_LAPLACE X  
 [ 1]'WHAT LEVEL OF SIGNIFICANCE DO YOU WISH TO TEST AT'  
 [ 2]'THE LEVEL SHOULD BE EXPRESSED AS A PERCENTAGE'  
 [ 3]A+

```

[ 4]A←NORMVALUE A
[ 5]→(X>A)/11
[ 6]→(X<-A)/12
[ 7]ΔA←0
[ 8]'NO TREND'
[ 9]→OPNOTD
[10]→0
[11]11:'TREND'
[12]'DETERIORATING TREND'
[13]ΔA←1
[14]OPTD
[15]→0
[16]12:'TREND IMPROVING'
[17]ΔA←1
[0]A←COMPARE_LIFETIME X
[1]RNOT ATTEMPTED YET
[0]A←COMPARE_LIKELIHOOD X
[1]RNOT ATTEMPTED YET
[0]A←COMPARE_PDF X
[1]RNOT ATTEMPTED YET
[ 0]A←COMPPDF X;V;V1;V2;V3;V4;V5
[ 1]RTHE AIM OF THIS FUNCTION IS TO COMPARE THE RESULTS OF DIFFERENT
[ 2]RLIKELIHOOD TESTS TO CHOOSE THE BEST OF THEM.
[ 3]V1←-2x+/(WEIBEST2 X)WIEBPDF X
[ 4]V2←-2x+/(NORMEST X)NORMPDF X
[ 5]V3←-2x+/(GAMMEST X)GAMMAPDF X
[ 6]V4←-2x+/(LOGNEST X)LOGNOPDF X
[ 7]V5←-2x+/(EXPOEST X)EXPOPDF X
[ 8]V←V1,V2,V3,V4,V5
[ 9]C←V√/V
[10]d←('WEIBULL')('NORMAL')('GAMMA')('LOGNORMAL')('EXPONENTIAL')
[11]A←d[C],' DISTRIBUTION'
[0]COMPTDLR

[1]R THE AIM OF THIS FUNCTION IS TO COMPARE THE RESULTS OF LEWIS - ROBINSON

```

[2]A TREND TEST WITH TABULAR VALUES

[3]→(Z=TABCHI)/a

[4]→(Z<TABCHI)/b

[5]b: 'THERE IS NO TREND'

[6]OPNOTD

[7]→0

[8]a: 'THERE IS TREND'

[9]OPTD

[ 0]A←COMP\_PDF X;V;V1;V2;V3;V4;V5

[ 1]A THE AIM OF THIS FUNCTION IS TO COMPARE THE RESULTS OF DIFFERENT

[ 2]A LIKELIHOOD TESTS TO CHOOSE THE BEST OF THEM.

[ 3]V1←-2x+/@(WEIBEST2 X)WIEBPDF X

[ 4]V2←-2x+/@(NORMEST X)NORMPDF X

[ 5]V3←-2x+/@(ΦGAMMEST X)GAMMAPDF X

[ 6]V4←-2x+/@(LOGNEST X)LOGNOPDF X

[ 7]V5←-2x+/@(÷EXPOEST X)EXPOPDF X

[ 8]V←V1,V2,V3,V4,V5

[ 9]C←V\|/V

[10]D←'WEIBULL','NORMAL','GAMMA','LOGNORMAL','EXPONENTIAL'

[11]A←D[C], ' DISTRIBUTION'

[0]A←X CORR Y

[1]A FN: TO CALCULATE CORRELATION COEFFICEINT BETWEEN TWO SERIES X AND Y

[2]A RA: A VECTOR OF NUMERIC VALUES

[3]A LA: A VECTOR OF NUMERIC VALUES THE SAME LENGTH AS TEH RA

[4]A RE: A SCALAR CORRELATION COEFFICIENT

[5]A

[6]MX←+/X÷ρX

[7]MY←+/Y÷ρY

[8]A←((+/X\*Y)-(ρX)×MX×MY)÷(((+/X\*2)-(ρX)×MX\*2)×((+/Y\*2)-(ρY)×MY\*2))\*0.5

```

[ 0]RES←CORRMAT MATRIX;R;RO;R2
[ 1]A FN: calculates correlation matrix
[ 2]A SS: iso/bs
[ 3]A RA: a matrix, rows of which correspond to observations
[ 4]A RA: and columns to the variables.
[ 5]A RE: a square matrix of correlations, the no. of cols the same as
[ 6]A RE: the no. of cols in right argument
[ 7]A
[ 8]A □ELX←'□ERROR ''ARGUMENT ERROR''
[ 9]R←MATRIX-ROρ(+≠MATRIX)÷1†RO←ρMATRIX
[10]R2←(+≠R*2)*0.5
[11]R2←R2◦.xR2
[12]RES←((≠R)+.xR)÷R2
[13]R2←~x≠MATRIX=(ρMATRIX)ρMATRIX[1;]
[14]RES←(R2◦.xR2)xRES
[ 0]RES←COVMAT MATRIX;R;RO
[ 1]A FN: calculates covariance matrix
[ 2]A SS: iso/bs
[ 3]A RA: a matrix, rows of which correspond to observations
[ 4]A RA: and columns to variables.
[ 5]A RE: a square matrix of covariances, the no. of cols the same as
[ 6]A RE: the no. of cols in the right argument
[ 7]A
[ 8]A □ELX←'□ERROR ''ARGUMENT ERROR''
[ 9]R←MATRIX-ROρ(+≠MATRIX)÷1†RO←ρMATRIX
[10]RES←((≠R)+.xR)÷~1+1†RO
[ 0]A←CO_LPTRD X
[1]A FN: TO COMPARE LAPLACE TEST STATISTIC TO THE NORMAL DISTRIBUTION
[2]A RA: TEST STATISTIC
[3]A RE: 1 + TREND : 0 NO TREND
[4]A
[5]A←NORMQ1 LEV
[6]A←(X≥A)+((-X)≥A)
[ 0]A←DF CO_MILTRD X

```

```

[1]AFN: TO COMPARE MIL-HDBK-189 TO CHI SQUARE DIST
[2]ARA: TEST STATISTIC
[3]ALA: NUMBER OF DEGREES OF FREEDOM
[4]ARE: 1 + TREND : 0 NO TREND
[5]A
[6]A←DF CHIQ1(1-LEV),LEV
[7]A←(X≤A[1])+(X≥A[2])
[0]A←CO_MILTRED X

[1]AFN: TO COMPARE MIL-HDBK-189 TO CHI SQUARE DIST
[2]ARA: TEST STATISTIC
[3]ALA: NUMBER OF DEGREES OF FREEDOM
[4]ARE: 1 + TREND : 0 NO TREND : 1 - TREND
[5]A
[6]A←DF CHI LEV
[7]A←(-X≤A[1])+(X≥A[2])
[0]A←CV X
[1]A←((VAR X)*0.5)÷MEAN X
[0]A←DATA

[1]A THE AIM OF THIS FUNCTION IS TO ASSIST THE USER MANIPULATE DATA
[2]'IS THE DATA SET A NEW SET (NEW) OR AN OLD SET (OLD)'
[3]→('NEW'=3+W)/a
[4]→('OLD'=W)/b
[5]a:DATARRAY
[6]→p
[7]b:OLDAT
[8]p:MENU
[ 0]DATARRAY
[ 1]INITIALD
[ 2]INITIAL_REPORT
[ 3]INITSTARTMEN
[ 4]DTWOALL
[ 5]DATVALID
[ 6]ENTERDAT X
[ 7]D←4 1,D

```



```

[ 8]X←ΔX
[ 9]ΔMEN1[1;2]←c 'DATA TESTED'
[10]START
[ 0]A←DATDEF
[ 1]D2←J←0
[ 2]D2a←' THIS IS THE INFORMATION WE HAVE ON DATA '
[ 3]D2←D2,c D2a
[ 4]R THE AIM OF THIS FUNCTION IS DEFINE DATA
[ 5]'DOES THE DATA CONSIST OF SINGLE OR MULTIPLE LIFETIMES ? S OR M'
[ 6]→('S'≠1↑)/m
[ 7]D2b←' SINGLE LIFETIMES '
[ 8]D2←D2,c D2b
[ 9]m:D2c←' MULTIPLE LIFETIMES '
[10]D2←D2,c D2c
[11]'ARE THERE ANY COVARIATES ? Y OR N'
[12]→('N'≠1↑)/c
[13]D2d←' NO COVARIATES '
[14]D2←D2,c D2d
[15]c:'ENTER NUMBER OF COVARIATES '
[16]f←
[17]D2e←f,' COVARIATES '
[18]D2←D2,c D2e
[19]'IS DATA DEPENDENT OR INDEPENDENT ? DEPENDENT / INDEPENDENT '
[20]→('D'≠1↑)/i
[21]D2f←' DATA IS DEPENDENT '
[22]D2←D2,c D2f
[23]i:D2g←' DATA IS INDEPENDENT'
[24]D2←D2,c D2g
[25]D[2]←c D2
[26]D[1]←c,'DATA IS NOW DEFINED'
[0]DATDEF1
[1]A←D[2]
[2]DATDEF1a
[3]DATDEF1d

```

```

[4]D[2]←c A
[ 0]DATDEF1a
[ 1]A←D[2]
[ 2]'IS THE DATA LIFETIME (LIFE) OR AGGREGATE DATA (AGG)'
[ 3]'I DO NOT KNOW (NOT)'
[ 4]→('L'=1+W1)/a
[ 5]→('AGG'=3+W)/b
[ 6]A[1;]←c '?'
[ 7]DATDEF1d
[ 8]b:A[1;]←c 'DATA IS AGGREGATE DATA'
[ 9]DATDEF1d
[10]a:A[1;]←c 'DATA IS LIFETIME DATA'
[11]'IS DATA A SET OF LIFETIMES (LIFE) OR'
[12]'IS DATA A SET OF FAILURE TIMES (FAILURE) '
[13]'I DO NOT KNOW (NOT)'
[14]→('L'=1+W1)/g
[15]→('F'=W)/h
[16]A[6;]←c '?'
[17]h:A[6;]←c 'DATA IS A SET OF FAILURE TIMES'
[18]→0
[19]g:A[6;]←c 'DATA IS A SET OF LIFETIMES'
[ 0]DATDEF1d
[ 1]'IS DATA A NUMBER OF GIVEN INTERVALS (INT) OR '
[ 2]'IS DATA A TOTAL AT GIVEN TIMES (TOT)'
[ 3]'IF YOU DON'T KNOW ENTER (NOT)'
[ 4]→('T'=W+11)/f
[ 5]→('I'=W)/i
[ 6]A[3;]←c '?'
[ 7]→0
[ 8]i:A[3;]←c 'DATA IS A GIVEN NUMBER OF INTERVALS'
[ 9]→0
[10]f:A[3;]←c 'DATA IS A TOTAL AT GIVEN TIMES'
[ 0]DATDEF2;A
[ 1]A←D[2]

```

```

[ 2]'IS DATA A SINGLE COMPONENT DATA (COMPONENT) OR'
[ 3]'IS DATA A SEQUENCE DATA (SEQUENCE)'
[ 4]→('C'=1↑□)/a
[ 5]A[2;]←c'DATA IS A SEQUENCE DATA'
[ 6]D[2]←cA
[ 7]→b
[ 8]a:A[2;]←c'DATA IS A SINGLE COMPONENT DATA'
[ 9]D[2]←cA
[10]→0
[11]b:'IS DATA A SINGLE SEQUENCE (SINGLE) OR '
[12]'IS DATA A MULTIPLE SEQUENCE (MULTIPLE)'
[13]→('SI'=1↑□)/c
[14]A[3;]←c'DATA IS A MULTIPLE SEQUENCE'
[15]D[2]←cA
[16]→0
[17]c:A[3;]←c'DATA IS A SINGLE SEQUENCE'
[18]D[2]←cA
[19]→0
[0]DATDEF4;A
[1]A←D[2]
[2]'IS DATA A COVARIATE DATA (COVARIATE) OR '
[3]'IS DATA A NO COVARIATE DATA (NO)'
[4]→('C'=1↑□)/a
[5]A[4;]←c'DATA IS A NO COVARIATE DATA'
[6]D[2]←cA
[7]→0
[8]a:A[4;]←c'DATA IA A COVARIATE DATA'
[9]D[2]←cA
[0]DATDEF5;A
[1]A←D[2]
[2]'IS DATA A CENSORED DATA (CENSORED) OR '
[3]'IS DATA NOT CENSORED DATA (NOT)'
[4]→('N'=1↑□)/a
[5]A[5;]←c'DATA IS A CENSORED DATA'

```

```

[6]D[2]←cA
[7]→0
[8]a:A[5;]←c'DATA IS NOT A CENSORED DATA'
[9]D[2]←cA
[0]A←DATSTRUCT
[1]␣THE AIM OF THIS FUNCTION IS TO ADD DATA TO THE ARRAY D
[2]' ENTER NAME OF DATA '
[3]s←␣
[4]S[1]←c s
[5]D[4]←c S
[ 0]A←DATVALID
[ 1]␣THE AIM OF THIS FUNCTION IS TO DESCRIBE THE CONTEXT OF DATA
[ 2]D3←␣J←0
[ 3]'WHAT WAS THE DATA COLLECTED ON ? '
[ 4]D3a←c␣
[ 5]'WHAT ARE THE COMPONENTS OF THE DATA ?'
[ 6]D3b←c␣
[ 7]'WHAT IS THE SERIAL NUMBER ?'
[ 8]D3c←c␣
[ 9]'WHERE WAS THE DATA COLLECTED ?'
[10]D3d←c␣
[11]'DOES DATA HAVE ANY PHYSICAL CONTEXT ? '
[12]D3e←c␣
[13]'ARE THERE ANY PECULIARITIES IN DATA '
[14]D3f←c␣
[15]D3←6 1␣D3a,D3b,D3c,D3d,D3e,D3f
[16]D[3]←cD3
[0]DECTDLR
[1]␣ THE AIM OF THIS FUNCTION IS TO RUN THE LEWIS - ROBINSON TREND TEST
[2]␣ ON DATA AND COMPARE THE RESULTS WITH THE TABULAR VALUE
[3]' PLEASE ENTER NAME OF DATA TO BE TESTED FOR TREND'
[4]W←␣
[5]Ⓢ'TRENDHM ',W
[6]COMPTDLR

```

```

[ 0] DESCRIBE; A; I; IO; R
[ 1] I ← IO - 1
[ 2] ABSTRACT
[ 3] L1:→ ('ABSTRACT' ^.=8+A+ (NL 3 4) [I+I+1;])/LA
[ 4]→ ('DESCRIBE' ^.=8+A)/LA
[ 5] R← CR A
[ 6] (←1+R[1;] \';')↑, R[1;]
[ 7] 79 ρ ^ '~'
[ 8] 0 1+ (←\((1 0+R)[;1]='R')) ≠ 1 0+R
[ 9] 0 ρ WA← DL 1.2
[10] LA:→ (I<1+ ρ NL 3)/L1

[0] C← DEVIDE A
[1] 'WHAT ARE THE POSITION OF THE VARIABLES TO BE YOUR TARGET VARIABLES ?'
[2] P← B+A[;P]
[3] ΔX1←B
[4] 'WHAT IS THE POSITION OF YOUR SORT VARIABLES ?'
[5] S←
[6] D←A[;S]
[7] ΔX←D
[0] A←DF LCENT
[1] R FN: FUNCTION TO CALCULATE THE LIKELIHOOD OF A SET OF CENSORED DATA
[2] R FN: HAVE A STUDENT t DISTRIBUTION
[3] R SS:
[4] R RA: A VECTOR OF LIFETIMES
[5] R LA: DEGREES OF FREEDOM
[6] R RE: SCALAR LIKELIHOOD OF CENSORED DATA HAVING t DISTRIBUTION
[7] R
[8] R
[9] A←(DF TDEN X)+(DF TCDF X)
[ 0] D←S DISPLAY A; IO; R; C; HL; HC; HT; HB; VL; VB; V; W; N; B

[ 1] R DISPLAY A GENERAL ARRAY IN PICTORIAL FORM
[ 2] R NORMAL CALL IS MONADIC. DYADIC CALL USED ONLY IN
[ 3] R RECURSION TO SPECIFY DISPLAY RANK, SHAPE, AND DEPTH.

```

[ 4 ]  $\square \text{IO} \leftarrow 0$

[ 5 ]  $\sharp (0 = \square \text{NC} \text{ 'S' }) / \text{'S'} \leftarrow \rho A$

[ 6 ]  $R \leftarrow \uparrow \rho, S$  A PSEUDO RANK.

[ 7 ]  $C \leftarrow \text{' } \ulcorner \sqcup \text{'}$  A UR, UL, LL, AND LR CORNERS.

[ 8 ]  $HL \leftarrow \text{'- '}$  A HORIZONTAL LINE.

[ 9 ]  $HC \leftarrow HL, \text{'@ '}, HL, \text{'~+@ '}$  A HORIZONTAL BORDERS.

[ 10 ]  $HT \leftarrow HC [ (0 < R) \times 1 + 0 < \uparrow^{-1} \uparrow, S ]$

[ 11 ]  $RW \leftarrow 0 \equiv \uparrow 0 \rho \subset (1 \uparrow \rho A) \uparrow A$

[ 12 ]  $HB \leftarrow HC [ 3 + 3 \lfloor (\text{'2x~A} \equiv \text{'A' } \square EA \text{ '1} + \uparrow \in 0 \times (1 \uparrow \times / \rho A) \uparrow, A') + 3 \times 1 < \rho \rho S ]$

[ 13 ]  $VL \leftarrow \text{'| '}$  A VERTICAL LINE.

[ 14 ]  $VB \leftarrow VL, \text{'\phi \downarrow '}$  A VERTICAL BORDER.

[ 15 ]  $V \leftarrow VB [ (1 < R) \times 1 + 0 < \uparrow^{-1} \uparrow, S ]$

[ 16 ]  $\sharp (0 \in \rho A) / \text{'A'} \leftarrow (1 \uparrow \rho A) \rho \subset \uparrow A$  A SHOW PROTOTYPE OF EMPTIES.

[ 17 ]  $\rightarrow (1 \leq A) / \text{GEN}$

[ 18 ]  $\rightarrow (2 < \rho \rho A) / D3$

[ 19 ]  $D \leftarrow \text{'A'}$  A SIMPLE ARRAYS.

[ 20 ]  $W \leftarrow 1 \uparrow \rho D \leftarrow (\text{'2} \uparrow 1 \text{ } 1, \rho D) \rho D$

[ 21 ]  $N \leftarrow \text{'1} \uparrow 1 \uparrow \rho D$

[ 22 ]  $\rightarrow (0 = \rho \rho A) / SS$

[ 23 ]  $D \leftarrow (C[1], V, ((W-1) \rho VL), C[2]), ((HT, N \rho HL), [0]D, [0]HB, N \rho HL), C[0], (W \rho VL), C[3]$

[ 24 ]  $\rightarrow 0$

[ 25 ]  $SS: HB \leftarrow ((0 \text{ ' '}) = \uparrow 0 \rho \subset A) / \text{'- '}$

[26]D←' ',(' ',[0]D,[0]HB,Nρ' '),','

[27]→0

[28]GEN:D←DISPLAY A R ENCLOSED ...

[29]N←Dv,≠' '

[30]D←(Nv~1φN)≠D

[31]D←(v≠' '∈D)/D

[32]D←((1,ρS)ρS)DISPLAY D

[33]→(2≥ρ,S)+D3E,0

[34]D3:D←0~1+0 1+≠c A R MULT-DIMENSIONAL ...

[35]W←1↑ρD

[36]N←~1+1↑ρD

[37]D←(C[1],V,((W-1)ρVL),C[2]),((HT,NρHL),[0]D,[0]HB,NρHL),C[0],(WρVL),C[3

[38]D3E:N←~2+ρ,S

[39]V←C[Nρ1],[0]VB[1+0<~2↑,S],[0](((~3+↑ρD),N)ρVL),[0]C[Nρ2]

[40]D←V,D

[ 0]DIST;W;a;b;K

[ 1]'WHICH OF THE FOLLOWING DISTRIBUTIONS ARE TO BE CONSIDERED NOW'

[ 2]NAMES

[ 3]'OR DO YOU WANT TO ADD ANY OTHER DISTRIBUTION (Y/N)'

[ 4]→('Y'=W←1↑□)/a

[ 5]→('N'=W)/b

[ 6]a:'ENTER THE NAME OF THE DISTRIBUTION YOU WANT TO ADD'

[ 7]K←□

[ 8]NAMES←NAMES,c K

[ 9]→0

[10]b:CHOOSDIST

[0]A←DISTCHILD X

```

[1]RPN: TO OBTAIN FROM DISTRIBUTION VECTOR THE CHILDREN COMPONENT
[2]RRA: DISTRIBUTION VECTOR NESTED VECTOR WITH 6 ELEMENTS
[3]RRA: 4TH ELEMENT THE CHILDREN MATRIX COLUMN 1 PARAMETERS
[4]RRA: COLUMN 2 DISTRIBUTIONS
[5]RRE: A VECTOR OF SCALAR NAMES
[6]A←(X[4])[;2]
  [0]A←DONE
[1]D1←J←0
[2]' THE AIM OF THIS FUNCTION IS TO FIND IF DATA IS DEFINED ON NOT '
[3]' IS DATA DEFINED ? Y OR N '
[4]→('Y'≠1+J)/a
[5]D1a←'DATA IS DEFINED'
[6]D1←D1,cD1a
[7]a:D1b←' DATA IS NOT DEFINED'
[8]D1←D1,cD1b
[9]D1
  [0]DTDEF1
[1]A←D[2]
[2]A[6;]←DATDEF1d
[3]D[2]←cA
  [ 0]A←DTHREE
[ 1]R THE AIM OF THIS FUNCTION IS TO DESCRIBE THE CONTEXT OF DATA
[ 2]D3←J←0
[ 3]'WHAT WAS THE DATA COLLECTED ON ? '
[ 4]D3a←
[ 5]'WHAT ARE THE COMPONENTS OF THE DATA ?'
[ 6]D3b←
[ 7]'WHAT IS THE SERIAL NUMBER ?'
[ 8]D3c←
[ 9]'WHERE WAS THE DATA COLLECTED ?'
[10]D3d←
[11]'DOES DATA HAVE ANY PHYSICAL CONTEXT ? '
[12]D3e←
[13]'ARE THERE ANY PECULIARITIES IN DATA '

```



```

[14]D3f←□
[15]D3←D3a,←D3b,←D3c,←D3d,←D3e,←D3f
[16]D[3]←←D3
  [ 0]A←DTWO
  [ 1]D2←←J←0
  [ 2]D2a←' THIS IS THE INFORMATION WE HAVE ON DATA '
  [ 3]D2←D2,←D2a
  [ 4]A THE AIM OF THIS FUNCTION IS DEFINE DATA
  [ 5]'DOES THE DATA CONSIST OF SINGLE OR MULTIPLE LIFETIMES ?  S OR M'
  [ 6]→('S'≠1↑□)/m
  [ 7]D2b←' SINGLE LIFETIMES '
  [ 8]D2←D2,←D2b
  [ 9]m:D2c←' MULTIPLE LIFETIMES '
[10]D2←D2,←D2c
[11]'ARE THERE ANY COVARIATES ? Y OR N'
[12]→('N'≠1↑□)/c
[13]D2d←' NO COVARIATES '
[14]D2←D2,←D2d
[15]c:'ENTER NUMBER OF COVARIATES '
[16]f←□
[17]D2e←f,' COVARIATES '
[18]D2←D2,←D2e
[19]'IS DATA DEPENDENT OR INDEPENDENT ? DEPENDENT / INDEPENDENT '
[20]→('D'≠1↑□)/i
[21]D2f←' DATA IS DEPENDENT '
[22]D2←D2,←D2f
[23]i:D2g←' DATA IS INDEPENDENT'
[24]D2←D2,←D2g
[25]D[2]←←D2
[26]D[1]←←,'DATA IS NOW DEFINED'
  [0]DTWOALL
  [1]DATDEF1
  [2]DATDEF2
  [3]DATDEF4

```

```

[4]DATDEF5
[5]D[1]←c, 'DATA IS NOW DEFINED'
[0]END

[1]R THE AIM OF THIS FUNCTION IS TO ENABLE THE USER TO QUIT THE SYSTEM:
[2]'DO YOU WANT TO SAVE THIS WORKSPACE (Y/N)'
[3]→('Y'=1+□)/1
[4]'THEN JUST ENTER )OFF '
[5]→0
[6]i:'ENTER )SAVE AND THEN ENTER )OFF'
[0]A←ENTERDAT X
[1]'ENTER DATA'
[2]ΔX←□
[3]R THE AIM OF THIS FUNCTION IS TO ENABLE THE USER ENTER NEW DATA
[4]R AND ADD DATA TO THE ARRAY D
[5]'ENTER NAME OF DATA'
[6]S←□
[7]D[4]←c S
[0]X ESTIMATION Y

[1]R THE AIM OF THIS FUCTION IS TO ESTIMATE THE PARAMETERS OF
[2]R A CHOSEN DISTRIBUTION
[3]NAMES
[4]d←□
[5]d←□
[0]A←EXPOEST X

[1]R FN: THIS FUNCTION ESTIMATES THE EXPONENTIAL DISTRIBUTION PARAMETERS.
[2]R LA: NOT AVAILABLE
[3]R RA: A VECTOR OF POSITIVE VALUES.
[4]R RS: ESTIMATES OF PARAMETRES.
[5]R SS:
[6]R AL: ALGORITHM ---- USING THE MEAN.
[7]A←+/X÷ρ X
[0]A←c EXPOLIKE X

[1]R FN: FUNCTION TO CALCULATE THE LOG LIKELIHOOD
[2]R FN: OF THE EXPONENTIAL DISTRIBUTION

```

```

[3]ARA: LIFETIME DATA
[4]ALA: MEAN OF EXPONENTIAL DISTRIBUTION
[5]ARE: A SCALER, LIKELIHOOD OF DISTRIBUTION
[6]A
[7]LE+
  [0]A←L EXPOPDF X
[1]A←Lx*-LxX
  [0]RES←DF FDEN X;Y;Z
[1]A FN: returns values of the F probability density function
[2]A SS: iso/bs
[3]A RA: scalar or vector of values at which probablities are required
[4]A LA: integer vector of length 2- degress of freedom of F distribution
[5]A RE: scalar or vector of probabilities of same length as right arg.
[6]A
[7]A ELX←'ERROR 'ARGUMENT ERROR''
[8]Y←(x/!Z-1)÷!+/-1,Z←0.5×DF
[9]RES←((x/DF×Z)×X*-1+0.5×DF[1])÷Y×(DF[2]+DF[1]×X)*0.5×+/DF
  [0]FITMODEL
  [0]A←FIT_DIST X
[1]AFN: FUNCTION TO FIT A DISTRIBUTION, CALCULATE PARAMETERS AND
[2]AFN: OBTIAN THE LOG LIKELIHOOD. COULD BE EXTENDED TO COVER
[3]AFN: ASSESSING FIT AND APPROPRIATE PLOTS.
[4]ARA: THE DISTRIBUTION TO BE FITTED.
[5]ARE: A NESTED VECTOR 1ST ELEMENT PARAMETERS OF DISTRIBUTION
[6]ARE: 2ND ELEMENT THE LOGLIKELIHOOD
[7]A
[8]A←A←',(>X[5]),D2
[9]A←A,A LOGLIKE X
  [ 0]A←FIT_DIST_ALL LIST
[ 1]AFN: TO FIT A SET OF DISTRIBUTION TO SET OF DATA
[ 2]ARA: A VECTOR OF NAMES OF DISTRIBUTION TO BE FITTED.
[ 3]ARE: A NESTED ARRAY OF NO OF DISTRIBUTION BY 2
[ 4]ARE: EACH ROW WILL CONSIST OF 2 ELEMENTS 1ST ELEMENT
[ 5]ARE: THE PARAMETERS OF DISTRIBUTION, 2ND ELEMENT THE LOGLIKELIHOOD

```

```

[ 6]ARE: OF FITTING THE DISTRIBUTION TO DATA.
[ 7]R
[ 8]J←0
[ 9]A←((P LIST),2)P' '
[10]L:A[J;]←FIT_DIST LIST[J+J+1]
[11]←(J<N)/L
[ 0]RES←DF FQUANT1 P0;Z;K1;K2;A1;A2;W;W1;W2;W3;W4;F;U
[ 1]R FN: Fast approximation to F Quantiles
[ 2]R RA: Scalar Probability at which quantile is required
[ 3]R LA: 2 item vector representing F degrees of freedom
[ 4]R RE: Scalar quantile
[ 5]R AL: NORMAL APPROX. BSC P.86
[ 6]R
[ 7]R ELX←'ERROR ' 'ARGUMENT ERROR' '
[ 8]ERR←1E-6
[ 9]FLAG←1
[10]←(ERR<|P0-0.5)/19
[11]RES←1
[12]←0
[13]19:←(P0>0.5)/18
[14]DF←1+DF
[15]P0←1-P0
[16]FLAG←0
[17]18:Z←NORMQUANT1 P0
[18]K1←1+DF
[19]K2←1+DF
[20]←(K2>3)/11
[21]U←Z÷K2*3÷4
[22]Z←(K2*3÷4)×+/1.1581 -0.2296 -0.0042 -0.0027×U*14
[23]11:A1←2÷9×K1
[24]A2←2÷9×K2
[25]W←Z×Z
[26]W1←1+A2×A2-W+2
[27]W2←A1+A2-1+A1×A2

```

```

[28]W3←1+A1×A1-W+2
[29]W4←((W2×W2)-W1×W3)÷2
[30]F←(W4-W2)÷W1
[31]RES←F*3
[32]→(FLAG)/0
[33]RES←÷RES
[ 0]RES←K FQUANT2 X;X1;X2;RES2;ERR;K1;K2;Z;U;A1;A2;W;W1;W2;W3;W4;F
[ 1]R FN: More accurate but slower than FQUANT1 - produces F quantiles
[ 2]R RA: scalar probability value for which quantile required
[ 3]R LA: 2 item vector degrees of freedom of F
[ 4]R RE: scalar F quantile
[ 5]R AL: iterations of FTAIL
[ 6]R
[ 7]R □ELX←'□ERROR ' 'ARGUMENT ERROR' '
[ 8]ERR←1E-6
[ 9]→(FLAG×X>0.5)/Z
[10]K←φK♦X←1-X
[11]Z:Z←NORMQUANT1 X
[12]K1←1+K
[13]K2←1+K
[14]→(K2>3)/11
[15]U←Z÷K2*3÷4
[16]Z←(K2*3÷4)×+/1.1581 -0.2296 -0.0042 -0.0027×U*1.4
[17]11:A1←2÷9×K1
[18]A2←2÷9×K2
[19]W←Z×Z
[20]W1←1+A2×A2-W+2
[21]W2←A1+A2-1+A1×A2
[22]W3←1+A1×A1-W+2
[23]W4←((W2×W2)-W1×W3)÷2
[24]F←(W4-W2)÷W1
[25]RES←F*3
[26]→(FLAG)/loop
[27]K←φK♦X←1-X♦RES←÷RES

```

```

[28]loop:X1←1-K FTAIL1 RES
[29]→(ERR≥|X1-X)/0
[30]RES2←RES×X÷X1
[31]X2←1-K FTAIL1 RES2
[32]RES←RES-(RES-RES2)×(X-X1)÷X2-X1↪loop
[ 0]RES←K FQUANT3 X;X1;X2;RES2;ERR
[ 1]R FN: F Quantiles
[ 2]R RA:Scalar Probability at which quantile is required
[ 3]R LA: 2 item vector representing F degrees of freedom
[ 4]R RE: Scalar quantile
[ 5]R AL: 1 ITERATION OF FQUANT1
[ 6]R
[ 7]R
[ 8]R ITERATIONS OF FTAIL
[ 9]ERR←1E-6
[10]RES←K FQAPP X
[11]loop:X1←1-K FTAIL RES
[12]→(ERR≥|X1-X)/0
[13]RES2←RES×X÷X1
[14]X2←1-K FTAIL RES2
[15]RES←RES-(RES-RES2)×(X-X1)÷X2-X1↪loop
[ 0]RES←DF FTAIL1 X;K1;K2;H1;H2;H3;G2;W1;W3;W4;W;I;P0;T1;NUM;DEN;T;N;MULT
;CON
[ 1]R FN: calculates the tail probabilities for the F distribution
[ 2]R FN: P(X>right argument)
[ 3]R RA: any shape - point to the right of which probabilities are calcul
ated
[ 4]R LA: degree of freedom (2- component vector - common to all RA value
s)
[ 5]R RE: An array of tail probs. same size as right hand argument.
[ 6]R AL: algorithm based on Majumder and Bhattacharjee (1873)
[ 7]R
[ 8]R □ELX←'□ERROR 'ARGUMENT ERROR''

```

```

[ 9]→(0≠ρρX)/nonint

[10]K1←1↑DF♦K2←1↓DF

[11]→(K1≠1)/cont1

[12]RES←2×K2 TTAIL X*÷2♦→0

[13]cont1:H1←K1÷2♦H2←K2÷2♦H3←H1+H2

[14]G2←1 1 -1+.*LOGGAMMA DF 2×H1,H2,H3

[15]W1←K2÷K2+K1×X

[16]W←H2≥W1×H3

[17]W3←1

[18]W4←W1÷1-W1

[19]H1←(W×H1)+(1-W)×H2

[20]I←⌊H1+H3×1-W1

[21]W1←(W×W1)+(1-W)×1-W1

[22]W4←W1÷1-W1

[23]I←⌊H1+H3×1-W1

[24]H2←(W×H2)+(1-W)×K1÷2

[25]N←10

[26]→((K1>19)∨(K1÷2)=⌊K1÷2)/cont

[27]CON←2/0 9 5 2,6ρ0

[28]MULT←2/0 3.63 1.917 1.177 0.7745 0.5188 0.3146 0.2 0.1 0.02

[29]N←N+⌈CON[K1]+MULT[K1]×K2

[30]cont:NUM←(W4×H1-W3+1+⌊I),(W1×H1-W3+I),W1×H3+0,⌊N

```

[31]DEN←H2+W3+<sup>-</sup>1+<sup>1</sup>I+N+2

[32]T←x\NUM÷DEN

[33]P0←1++/T

[34]W1←\*((H2×W1)+(H1-1)×\*(1-W1))-G2)

[35]RES←P0×W1÷H2

[36]RES←(W×RES)+(1-W)×1-RES♦→0

[37]nonint:W1←<sup>ρ</sup>X♦X←,X♦N←<sup>ρ</sup>X♦I←1♦RES←<sup>1</sup>0

[38]loop:RES←RES,DF FTAIL1 X[I]

[39]→(N≥I←I+1)/loop

[40]RES←W1<sup>ρ</sup>RES

[ 0]RES←DEG FTAIL2 Y;X;A;DEGAV;TM;S;TP;TM;B;PAR;E;G;I;J;R

[ 1]A FN: calculates the tail probabilities for the F distribution

[ 2]A FN: P(X>right argument)

[ 3]A SS: iso/bs

lated [ 4]A RA: any shape - point to the right of which probabilities are calcu

t [ 5]A RE: An array of tail probs of F dist same size as right hand argumen

[ 6]A AL: algorithm based on Majumder and Bhattacharjee (1873)

s) [ 7]A LA: degree of freedom (2- component vector - common to all RA value

[ 8]A

[ 9]A □ELX←'□ERROR ''ARGUMENT ERROR''

[10]X←DEG[2]÷DEG[2]+DEG[1]×Y

[11]DEGAV←+/DEG←DEG÷2

[12]DEG←DEG,[0.5]X,1-X



```

[13] DEG ← (DEG[1;2] < DEGAV × X) ⊕ DEG
[14] TM ← A + B + 1
[15] S ← [DEG[1;1] + DEG[2;2] × DEGAV
[16] con: TP ← DEG[1;1] - A
[17] R ← (÷ \DEG[2;]) [1 + S ≠ 0]
[18] con2: B ← B + TM - TM × TP × R ÷ DEG[1;2] + A
[19] TP ← | TM
[20] → (TP ≤ 1E-7 × B[1]) † con3
[21] A ← A + 1
[22] → (0 < S + S - 1) † con
[23] DEGAV ← 1 + TP - DEGAV
[24] → con2
[25] con3:
[26] PAR ← 2 × (+ / DEG[1;]), DEG[1;]
[27] RES ← \ 0
[28] I ← 0
[29] loop:
[30] → (3 < I - I + 1) † endloop
[31] S ← 0
[32] J ← 2 | PAR[I]
[33] → (PAR[I] ≤ 2) † con4
[34] S ← + / * ( - 0.5 × J) + \ L (PAR[I] - 1) ÷ 2
[35] con4: RES ← RES, S + J × 0.57236494

```

[36]→loop

[37]endloop:

[38]E←RES

[39]G←RES[1]-RES[2]+RES[3]

[40]B←B×\*G+(DEG[1;2]×\*DEG[2;1])+(DEG[1;1]-1)×\*DEG[2;2]

[41]B←B÷DEG[1;2]

[42]RES←(1-B,B)[1+DEG[1;2]≥DEG[2;1]×DEGAV]

[0]RES←S\_SC GAMMADEN X;Y;Z

[1]R FN: returns values of the gamma probability density function

[2]R SS: iso/bs

[3]R RA: scalar or vector of values at which probabilities are required

[4]R LA: vector of length 2, 1st element is shape, 2nd scale.

[5]R RE: scalar or vector of probabilities of same length as right argumen

[6]R AL: ?

[7]R

[8]R ELX←'ERROR 'ARGUMENT ERROR''

[9]RES←(X\*Z)×(\*-Y×X)÷(!Z←S\_SC[1]-1)÷(Y←S\_SC[2])\*S\_SC[1]

[0]A←GAMMALIKE X

[1]R FN: FUNCTION TO CALCULATE THE LOG LIKELIHOOD

[2]R FN: OF THE GAMMA DISTRIBUTION

[3]R RA: LIFETIME DATA

[4]R LA: THE PARAMETERS FOR THE GAMMA DISTRIBUTION

[5]R RE: A SCALER VALUE, THE LIKELIHOOD

[6]R

[7]La←+/k\*a+(+/(k-1)\*X)-(+/aX)+(ρX-1)!k+(+/(1-I)

```

[8]R I←BOTTOM SHELF
[0]A←a_k GAMMAPDF T
[1]a←a_k[1]
[2]k←a_k[2]
[3]A←a×((a×T)×(k-1))×(*-a×T)÷!k
[0]A←GAMMEST X
[1]R FN: THIS FUNCTION ESTIMATES THE GAMMA DISTRIBUTION PARAMETERS.
[2]R LA: NOT AVAILABLE .
[3]R RA: VECTOR OF POSITIVE VALUES.
[4]R RS: ESTIMATES OF PARAMETERS OF THE GAMMA DISTRIBUTION.
[5]R SS:
[6]R AL: ALGORITHM---USING THE SAMPLE MEAN AND SAMPLE VARIANCE
[7]R b = SCALE PARAMETRE
[8]R c = SHAPE PARAMETRE
[9]m←+/X÷ρX
[10]s←(((ρX)+.×X)-(ρX)×m÷.×m)÷-1+1÷ρX
[11]b←s÷m
[12]c←(m÷s)*2
[13]A←(c b),(c c)
[0]GUIDANCE

```

```

STEM [1]R FN : TO GUIDE THE USER AND SUGGEST FURTHER ANALYSIS STEPS IN USING SY
[2]R
[3]R
[4]R
[5]R
[6]R
[7]ΔMEN1
[8]'DECIDE ON PATH OF ENQUIRY BY CHOOSING ONE OF THE ABOVE SYSTEM TASKS'
[9]t←□

```

[10]→(t=1)/i

[11]i:ΔMEN2

[ 0]HELP

[ 1]' THE AIM OF THIS FUNCTION IS TO PROVIDE HELP FACILITIES TO ASSIST'

[ 2]' THE USER IN USING ARDA. HELP IS DEVIDED IN FOUR CATEGORIES :'

[ 3]' 1- KNOWLEDGE, 2- GUIDANCE, 3- EXPLANATION AND 4- DESCRIBE'

[ 4]' TO CHOOSE ANY FACILITY JUST TYPE ITS NUMBER AND PRESS RETURN '

[ 5]R

[ 6]H←□

[ 7]→(H=1)/H1

[ 8]→(H=2)/H2

[ 9]→(H=3)/H3

[10]→(H=4)/H4

[11]H1:KNOWLEDGE

[12]→0

[13]H2:GUIDANCE

[14]→0

[15]H3:EXPLANATION

[16]→0

[17]H4:DESCRIBE

[18]→0

[0]A←HPPFIT X

[1]RPN: TO FIT HPP TO FAILURE TIME DATA

[2]RPA: VECTOR OF FAILURE TIMES,  $T_i \geq T_{i-1}$ .

[3]RPR: A SCALAR RATE OF OCCURRENCE

[4]R

[5]'FITTING EXPONENTIAL'

[6]A←(1-X)/X

[0]A←IN A

[0]INITDATA

[1]DATA←4ρ ' '

[2]DATA[1]←c 'DATA IS NOT DEFINED'

[3]DATA[2]←c 6 1ρ ' '

```

[0]INITIALD
[1]D←4ρ ' '
[2]D[1]←c 'DATA IS NOT DEFINED'
[3]D[2]←c 6 1ρ ' '
[0]A← INITIALIZE
[1]REPORT← (5 1)ρ←ρ ' '
[ 0]INITIALMEN2
[ 1]ΔMEN2[1;1]←c '          1 - TREND'
[ 2]ΔMEN2[2;1]←c '          2 - INDEPENDENCE'
[ 3]ΔMEN2[3;1]←c '          3 - PDF'
[ 4]ΔMEN2[4;1]←c '          4 - LIKELIHOOD'
[ 5]ΔMEN2[5;1]←c '          5 - LIFETIMES'
[ 6]ΔMEN2[6;1]←c '          6 - MAIN MENU'
[ 7]ΔMEN2[1;2]←c '          NOT TESTED'
[ 8]ΔMEN2[2;2]←c '          NOT TESTED'
[ 9]ΔMEN2[3;2]←c '          NOT TESTED'
[10]ΔMEN2[4;2]←c '          NOT TESTED'
[11]ΔMEN2[5;2]←c '          NOT TESTED'
[ 0]INITIALMENU
[ 1]MEN1a←5 1ρ ' '
[ 2]MEN1a[1;]←c '1 - DATA ANALYSIS'
[ 3]MEN1a[2;]←c '2 - OBJECTIVES ANALYSIS'
[ 4]MEN1a[3;]←c '3 - MODELS ANALYSIS'
[ 5]MEN1a[4;]←c '4 - STATISTICAL KNOWLEDGE'
[ 6]MEN1a[5;]←c '5 - QUIT'
[ 7]MEN1b←5 1ρ ' '
[ 8]MEN1b[1;]←c '          NOT TESTED'
[ 9]MEN1b[2;]←c '          NOT TESTED'
[10]MEN1b[3;]←c '          NOT TESTED'
[11]MEN1b[4;]←c '          NOT ACCESSED'
[12]MEN1b[5;]←c '          NOT TESTED'
[13]MEN1←MEN1a,MEN1b
[0]INITIAL_REPORT
[1]INITIALIZE

```

```

[2]REPORT[1;]←c ' THE REPORTE IS EMPTY SINCE NO TEST HAS BEEN DONE YET'
[0]INITMOD
[1]MOD←3ρ ' '
[2]MOD[1]←c 'MODEL IS NOT DEFINED'
[0]INITOBJ
[1]OBJ←2ρ ' '
[2]OBJ[2]←c 'OBJECT IS NOT DEFINED'
[0]INITSTARTMEN
[1]ΔMEN1[1;2]←c 'NOT TESTED'
[2]ΔMEN1[2;2]←c 'NOT TESTED'
[3]ΔMEN1[3;2]←c 'NOT TESTED'
[4]ΔMEN1[4;2]←c 'NOT TESTED'
[0]INITSYSTEM
[1]INITDATA
[2]INITMOD
[3]INITOBJ
[4]LOG← ' '
[5]REPORT← ' '
[0]A←IS_IT Y
[1]LOG←X[2]
[2]L:± 'A← ' , , ⊃ LOG[1;]
[3]→ (A=1)/K
[4]'SINCE', LOG[;1], 'DOES NOT HOLD. STOP QUERY'
[5]→ 0
[6]K:→ (1=1+ρ LOG)/0
[7]LOG←1 0+LOG
[8]→ L
[0]A←IS_THERE X
[1]LOG←X[2]
[2]L:± 'A← ' , , ⊃ LOG[1;]
[3]→ (A=1)/K
[4]'SINCE ' , ( , ( ⊃ 1 1+REPORT) ) [1] , ' . STOP QUERY'
[5]→ 0
[6]K:→ (1=1+ρ LOG)/0

```

[7]LOG+1 0+LOG

[8]+L

[0]KNOWLEDGE

[1]' THE AIM OF THIS FUNCTION IS TO PROVIDE A POOL OF INFORMATION '

[2]' ON SUBJECTS RELATED TO ARDA AND TO RELIABILITY DATA ANALYSIS '

[3]' TO ACCESS KNOWLEDGE ON ANY OF THE TOPICS BELOW JUST ENTER THE '

[4]' ITS NUMBER AND PRESS RETURN'

[5]'1- OBJECTIVES, 2- MODEL, 3-DATA, 4-NORMAL DIST., 5- GAMMA DIST.,'

[6]'6- WEIBALL DIST. , 7-EXPONENTIAL DIST. , 8- LOG NORMAL DIST. ,'

[7]'9- NHPP , 10- BPP , 11- HPP , 12- RENEWAL PROCESS'

[0]A+LCENCHI

[1]R FN: FUNCTION TO CALCULATE THE LIKELIHOOD OF A SET OF CENSORED DATA

[2]R FN: TO BE CHI SQUARED

[3]R SS:

[4]R RA: A VECTOR OF LIKE TIMES

[5]R LA:

[6]R RE: SCALER LIKELIHOOD OF BISTRIBUTION BEING CHI SQUARED

[7]R

[8]R

[9]A+@ (CHIPDF ΔCENSOR/X)+@ (CHICDFM(ΔCENSOR)/X)

[ 0]A+LCENNORM X;a;b;CV;J;P

EING

[ 1]R FN: FUNCTION TO CALCULATE THE LIKELIHOOD OF A SET OF CENSORED DATA B

[ 2]R FN: OF THE NORMAL DISRTIBUTION

[ 3]R SS:

[ 4]R RA: A VECTOR OF LIFETIMES

[ 5]R LA:

[ 6]R RE: SCALER LIKELIHOOD OF CENSORED DATA BEING OF NORMAL DISTRIBUTION.

[ 7]R

[ 8]R

```

[ 9]J←0

[10]A←9#0

[11]a_b←NORMEST X

[12]a1_b1←MATRIX a_b

[13]loop:CV←a1_b1[J+J+1]

[14]A[J]←(+/#(CV NORMPDF ΔCENSOR/X))+(+/#(1-(CV NORMCDF(ΔCENSOR)/X)))

[15]→(9>J)/loop

[16]P←A\ (Γ/A)

[17]A←A[P],a1_b1[P]

```

```

[ 0]A←DF LCENT X

```

```

[1]R FN: FUNCTION TO CALCULATE THE LIKELIHOOD OF A SET OF CENSORED DATA

```

```

[2]R FN: HAVE A STUDENT t DISTRIBUTION

```

```

[3]R SS:

```

```

[4]R RA: A VECTOR OF LIFETIMES

```

```

[5]R LA:

```

```

[6]R RE: SCALER LIKELIHOOD OF CENSORED DATA HAVING t DISTRIBUTION

```

```

[7]R

```

```

[8]R

```

```

[9]A←(#(DF TDEN ΔCENSOR/X))+(#(1-(DF TCDF(ΔCENSOR)/X)))

```

```

[ 0]A←LCENWEIB X

```

```

[ 1]R FN: FUNCTION TO CALCULATE THE LIKELIHOOD OF A SET OF CENSORED DATA B
EING

```

```

[ 2]R FN: OF THE WEIBULL DISRTIBUTION

```

```

[ 3]R SS:

```

```

[ 4]R RA: A VECTOR OF LIFETIMES

```

```

[ 5]R LA: a = WEIBULL SCALE PARAMETER, b = WEIBULL SHAPE PARAMETER

```

```

[ 6]R RE: SCALER LIKELIHOOD OF CENSORED DATA BEING OF WEIBULL DISTRIBUTION

```



[ 7]R

[ 8]R

[ 9]a\_b←WEIBEST2 X

[10]MATRIX a\_b

[11]A←(+/(a1\_b1 WEIBPDF ΔCENSOR/X))+(+/(1-((a1\_b1)WEIBCDF(ΔCENSOR)/X))

[0]A←LIFE X

[1]→(0=ρX)/b

[2]'DATA HAS MULTIPLE SEQUENCES'

[3]→c

[4]b:'DATA IS SINGLE LIFETIMES OR SEQUENCE OF LIFETIMES'

[5]c:Z→X

[0]A←LIFETIMES

[1]' ENTER THE NAME OF YOUR DATA ARRAY'

[2]XX←□

[3]A←(1+XX)<sup>-1</sup>+XX

[4]B←ρXX

[5]RA←B 1ρA

[6]'LIFETIMES OF THIS DATA ARE ',A

[7]→(B≠0)/b

[8]' DATA CONSISTS OF SINGLE LIFETIMES'

[9]b:'DATA CONSISTS OF MULTIPLE SEQUENCES'

[0]RES←LOGGAMMA W;IND

[1]R FN: calculated the log gamma function of W-1

[2]R RA: any shape, positive value -argument of log gamma function

[3]R RE: An array of log gamma values same size as right hand argument

[4]R

[5]R DELX←'ERROR ' 'ARGUMENT ERROR''

[6]IND←⌊W-1

[7]RES←(⌊1+W)++/((W-1)°≥IND)×⌊-(1+W)°.-IND

[ 0]RES←LOGGAMMA DF;S;J

[ 1]R FN: Calculates the log of the gamma function of -1+0.5×DF

[ 2]R RA: a vector of positive values

```

[ 3]R RE: vector of loggamma values same lenght as right argument
[ 4]R AL: faster than loggamma with larger range
[ 5]R
[ 6]R DELX←'ERROR ' 'ARGUMENT ERROR'
[ 7]←(DF<343)+loop1♦RES←*!~1+DF÷2♦+0 R Numeric limit
[ 8]loop1:J+2|DF
[ 9]S←+/(~0.5×J)+\L (DF-1)÷2
[10]RES←S+*!~0.5×J
[0]A←PAR LOGLIKE X
[1]R FN: FUNCTION TO CALCULATE THE LOGLIKELIHOOD OF A SET OF
[2]R FN: DATA FOR A GIVEN DISTRIBUTION
[3]R RA: THE DISTRIBUTION FOR WHICH THE LIKELIHOOD REQUIRED
[4]R LA: THE PAR OF THE DISTRIBUTION
[5]R RE: A SCALAR VALUE, THE LOGLIKELIHOOD
[6]R
[7]A←+/(PAR X[5](CENA)/D2),(PAR X[6](CENL)/D2),PAR X[7](CENR)/D2
[0]A←LOGNEST X
[1]R THIS FUNCTION ESTIMATES THE LOGNORMAL DISTRIBUTION PARAMETERS.
[2]R LA: NOT AVAILABLE.
[3]R RA: VECTOR OF POSITIVE VALUES.
[4]R RS: ESTIMATES OF THE PARAMETERS OF THE LOGNORMAL DISTRIBUTION.
[5]R SS:
[6]R AL: ALGORITHM--- USING THE MEAN OF LOG (L) u.
[7]u←(+/(X)÷ρX
[8]a←(+/(X-u)*2))÷~1+ρX
[9]A←a,u
[0]A←a_q LOGNOPDF X
[1]a←a_q[1]
[2]q←a_q[2]
[3]A←+/(~(a*0.5)-0.5×(÷a)×(X-q)*2)
[ 0]al_b1←MATRIX a_b;a;b;al_b1
[ 1]R FN: THIS FUNCTION CALCULATES THE VALUES NEAR TO PARAMETERS
[ 2]R FN:
[ 3]R SS:

```

```

[ 4]R RA: DISTRIBUTION ESTIMATED PARAMETERS
[ 5]R LA:
[ 6]R RE: ARRAY OF ESTIMATED PARAMETERS
[ 7]R
[ 8]a←a_b[1]
[ 9]b←a_b[2]
[10]a1_b1←9ρ ' '
[11]a1_b1[1]←c (a÷2) , (b÷2)×3
[12]a1_b1[2]←c (a÷2) , b
[13]a1_b1[4]←c a , (b÷2)×3
[14]a1_b1[5]←c a , b
[15]a1_b1[6]←c a , b÷2
[16]a1_b1[7]←c ((a÷2)×3) , (b÷2)×3
[17]a1_b1[8]←c ((a÷2)×3) , b
[18]a1_b1[9]←c ((a÷2)×3) , b÷2
[19]a1_b1[3]←c (a÷2) , b÷2
[ 0]a2_b2←MATRIX2 a_b
[ 1]a←a_b[1]
[ 2]b←a_b[2]
[ 3]a2_b2←9ρ ' '
[ 4]a2_b2[1]←c (a×3)÷4 , (b×5)÷4
[ 5]a2_b2[2]←c a , (b×5)÷4
[ 6]a2_b2[3]←c (a×5)÷4 , (b×5)÷4
[ 7]a2_b2[4]←c (a×3)÷4 , b
[ 8]a2_b2[5]←c a , b
[ 9]a2_b2[6]←c (a×5)÷4 , b
[10]a2_b2[7]←c (a×3)÷4 , (b×3)÷4
[11]a2_b2[8]←c a , (b×3)÷4
[12]a2_b2[9]←c (a×5)÷4 , (b×3)÷4
[ 0]a1_b1←dist MATRIX3 a_b;a;b;a1_b1
[ 1]R FN: THIS FUNCTION CALCULATES THE VALUES NEAR TO PARAMETERS
[ 2]R FN:
[ 3]R SS:
[ 4]R RA: DISTRIBUTION ESTIMATED PARAMETERS

```

```

[ 5]A LA:
[ 6]A RE: ARRAY OF ESTIMATED PARAMETERS
[ 7]A
[ 8]a←a_b[1]
[ 9]b←a_b[2]
[10]a1_b1←9, ' '
[11]a1_b1[1]←c (a-dist[1]), (b+dist[2])
[12]a1_b1[2]←c (a-dist[1]), b
[13]a1_b1[4]←c a, (b+dist[2])
[14]a1_b1[5]←c a, b
[15]a1_b1[6]←c a, b-dist[2]
[16]a1_b1[7]←c (a+dist[1]), (b+dist[2])
[17]a1_b1[8]←c (a+dist[1]), b
[18]a1_b1[9]←c (a+dist[1]), b-dist[2]
[19]a1_b1[3]←c (a-dist[1]), b-dist[2]
[ 0]RES←MAX ARRAY
[ 1]A FN: calculates maximum of a set of values
[ 2]A SS: iso/bs
[ 3]A RA: array of values for which maximum is to be calculated
[ 4]A RE: maximum across the first dimension of the right argument,
[ 5]A RE: the dimensions are those of the
[ 6]A RE: right argument minus the first dimension
[ 7]A
[ 8]A □ELX←'□ERROR ' 'ARGUMENT ERROR''
[ 9]RES←[≠ARRAY
[10]
[0]RES←MEAN ARRAY
[1]A FN: calculates mean of a set of values
[2]A SS: iso/bs
[3]A RA: array of values for which mean is to be calculated
[4]A RE: mean across the first dimension of the right argument,
[5]A RE: the dimensions are those of the
[6]A RE: right argument minus the first dimension
[7]A

```

```

[8]A DELX←'ERROR 'ARGUMENT ERROR''
[9]RES←(÷ARRAY)÷1÷ARRAY÷1/ARRAY
[0]RES←MEANFT FTAB
[1]A FN: calculates mean of a frequency table
[2]A SS: iso/bs
[3]A RA: matrix where first column is X value and
[4]A RA: second column is frequency
[5]A RE: scalar mean
[6]A
[7]A DELX←'ERROR 'ARGUMENT ERROR''
[8]RES←(÷X/FTAB)÷(÷FTAB)[2]
[0]RES←MEDIAN X;T;B
[1]A FN: calculates median of set of values
[2]A SS: iso/bs
[3]A RA: vector of values for which median is to be calculated
[4]A RE: scalar median
[5]A
[6]A DELX←'ERROR 'ARGUMENT ERROR''
[7]RES←0.5×÷(X[÷X])[÷0.5×0 1÷X]
[0]A←MENU
[1]A THE AIM OF THIS FUNCTION IS TO DECIDE ON THE PATH OF ANALYSIS
[2]A THE USER WANTS.
[3]'THE FOLLOWING IS A SET OF OPTIONS AVAILABLE IN THE SYSTEM TO USE'
[4]' '
[5]'YOU CAN DECIDE OF THE PATH OF ANALYSIS YOU REQUIRE BY TYPING THE '
[6]'NAME OF THE FUNCTION SHOWN NEXT TO YOUR CHOICE OF ACTIVITY'
[7]' '
[8]ΔΔMENU1
[9]ΔMEN1
[10]A←COLLECT '12345'
[11]ΔΔ←'START',A
[0]MENU1a
[1]÷('1'=1÷Δc)/d
[2]÷('2'=1÷Δc)/o

```

```

[ 3]→('3'=1†Δc)/m
[ 4]→('4'=1†Δc)/k
[ 5]→('5'=1†Δc)/e
[ 6]d:DATA
[ 7]MEN1b[1;]←c '          DATA TESTED'
[ 8]→p
[ 9]o:OBJECTIVES
[10]MEN1b[2;]←c '          OBJECTIVES TESTED'
[11]→p
[12]m:MODELS
[13]MEN1b[3;]←c '          MODELS TESTED'
[14]→p
[15]k:KNOWLEDGE
[16]MEN1b[4;]←c '          KNOWLEDGE ACCESSED'
[17]p:MENU
[18]e:END
[ 0]MENU2
[1]'WOULD YOU LIKE TO TEST DATA FOR '
[2]ΔMEN2
[3]' '
[4]TESTS
[ 0]MENU3a
[ 1]□←s
[ 2]→('1'=1†s)/i
[ 3]→('2'=1†s)/m
[ 4]→('3'=1†s)/a
[ 5]→('4'=1†s)/e
[ 6]i:TEST_INDEPENDENCE
[ 7]→p
[ 8]m:MENUE
[ 9]→p
[10]a:USER_DESIRE
[11]p:MENUE2
[12]e:END

```

```

[ 0]MENU3b
[ 1]v←□
[ 2]→('1'=1†v)/n
[ 3]→('2'=1†v)/c
[ 4]→('3'=1†v)/m
[ 5]→('5'=1†v)/e
[ 6]n:NHPP
[ 7]→p
[ 8]c:NONSTATMOD
[ 9]→p
[10]m:MENU
[11]→p
[12]a:USER_DESRIE
[13]p:MENU2
[14]→0
[15]e:END

```

```

[0]RES←MIDPOINTS VEC;U
[1]A FN: calculates midpoints of increasing left-hand endpoints
[2]A SS: iso/bs
[3]A RA: vector of endpoints
[4]A RE: vector of midpoints
[5]A
[6]A □ELX←'□ERROR ''ARGUMENT ERROR''
[7]RES←1†(−1 2+.x−2†U),U←0.5x−1†VEC+1†VEC

```

```

[ 0]RES←MIN ARRAY
[ 1]A FN: calculates minimum of a set of values
[ 2]A SS: iso/bs
[ 3]A RA: array of values for which minimum is to be calculated
[ 4]A RE: minimum across the first dimension of the right argument,
[ 5]A RE: the dimensions are those of the
[ 6]A RE: right argument minus the first dimension
[ 7]A
[ 8]A □ELX←'□ERROR ''ARGUMENT ERROR''
[ 9]RES←[≠ARRAY

```

```

[10]
  [ 0]A←MLCENNORM X
[ 1]C←2ρ ' '
[ 2]K←0
[ 3]A←9ρ0
[ 4]C←LCENNORM X
[ 5]□←dist←0.45×C[2]
[ 6]loop2:a1_b1←dist MATRIX3=C[2]
[ 7]B1←C[1]
[ 8]K←K+1
[ 9]FRED←a1_b1
[10]FRED[(FRED[;1]≤0)/^9;1]←10
[11]a1_b1←C[2]FRED
[12]J←0
[13]loop1:CV←a1_b1[J+J+1]
[14]A[J]←(+/#(CV NORMPDF ΔCENSOR/X))+(+/#(1-(CV NORMCDF(ΔCENSOR)/X)))
[15]→(9>J)/loop1
[16]P←A\ (r/A)
[17]B←A[P]
[18]□←C+A[P],a1_b1[P]
[19]→(0.001<|B1-B)/loop2
[20]dist←dist×0.5
[21]C
[22]→(K<16)/loop2
[23]A←C
  [ 0]A←MLCENNORM2 X
[ 1]C←2ρ ' '
[ 2]K←0
[ 3]A←9ρ0
[ 4]C←LCENNORM X
[ 5]□←dist←0.45×C[2]
[ 6]loop2:a1_b1←dist MATRIX3=C[2]
[ 7]B1←C[1]
[ 8]K←K+1

```



```

[ 9]FRED←a1_b1
[10]FRED[(FRED[;1]≤0)/∧9;1]←10
[11]a1_b1←c[2]FRED
[12]J←0
[13]loop1:CV←a1_b1[J+J+1]
[14]V1←1-(CV NORMCDF(ΔCENSOR)/X)
[15]V2←CV NORMCDF ΔCENSOR/X
[16]A[J]←(+/#(V1≥1E-6)/V1)-((+/V1<1E-6)×1000)
[17]A[J]←A[J]←(+/#(V2≥1E-6)/V2)-((+/V2<1E-6)×1000)
[18]A[J]←(+/#(V<1E-6)/V)-((+/V<1E-6)×1000)
[19]A[J]←(+/#(CV NORMPDF ΔCENSOR/X))+(+/#(1-(CV NORMCDF(ΔCENSOR)/X)))
[20]→(9>J)/loop1
[21]P←A∧(Γ/A)
[22]B←A[P]
[23]B←C←A[P],a1_b1[P]
[24]→(0.001<|B1-B)/loop2
[25]dist←dist×0.5
[26]C
[27]→(K<30)/loop2
[28]A←C
[ 0]RES←MODE DATA;I;T;Z
[ 1]A FN: calculates mode of set of values
[ 2]A SS: iso/bs
[ 3]A RA: vector of values for which mode is to be calculated
[ 4]A RE: scalar mode (or vector if more than 1 mode)
[ 5]A
[ 6]A ELX←'ERROR ' 'ARGUMENT ERROR'
[ 7]RES←∧0
[ 8]DATA←,DATA
[ 9]I←1,1←DATA≠1←DATA←DATA[4DATA]
[10]T←I/∧I
[11]A check if any repeated values
[12]→((ρT)=ρDATA)←0
[13]DATA←I/DATA

```

```

[14] RES← ((I/Z)=Z←((1+T), 1+ρ I)-T)/DATA
[15]
[0] A←MODELS
[1] A THE AIM OF THIS FUNCTION IS TO ASSIST THE USER IN DECIDING ON THE
[2] A APPROPRIATE MODEL FOR DATA.
[3] 'ENTER NAME OF DATA'
[4] C←
[5] COMP_PDF C
[0] RES←POWER MOMENT ARRAY
[1] A FN: calculates a moment about the origin
[2] A SS: iso/bs
[3] A RA: array of values for which moment is required
[4] A LA: scalar moment value required
[5] A RE: moment, dimensions are those of right argument
[6] A RE: minus first dimension
[7] A
[8] A DELX←'ERROR ' 'ARGUMENT ERROR''
[9] RES← (+ARRAY*POWER) ÷ 1+ρ ARRAY← 1/ARRAY
[ 0] A←NEXT_ACTION B
[ 1] → (B=1)/L1
[ 2] 'SINCE LAST ACTION RESULTED IN NEGATIVE STOP QUERY'
[ 3] A←B
[ 4] → 0
[ 5] L1:→ (1=1+ρ LOG)/L3
[ 6] LOG← 1 0+LOG
[ 7] 'SUGGESTION IS ', , , LOG[1;]
[ 8] 'DO YOU WANT TO RUN IT (Y/N)?'
[ 9] → (Y=COLLECT1 'YN')/L2
[10] → 0
[11] L2:← 'A+', , , LOG[1;]
[12] L3: 'END OF SEARCH'
[13] A←B
[0] NHPP
[1] 'NOT YET IMPLEMENTED'

```

```

[0]A←NHPPFIT T
[1]AFN: TO ESTIMATE PARAMETERS OF CROWS NHPP MODEL BETA LAMBDA
[2]ARA: VECTOR OF FAILURE TIMES,  $T_i \geq T_{i-1}$ 
[3]ARE: A TWO ELEMENT VECTOR BETA LAMBDA
[4]ASS: ASCHER AND FEINGOLD (P84)
[5]A
[6]A←( $\rho T$ )÷+/ $\#$ ( $-1 \uparrow T$ )÷ $-1 \uparrow T$ 
[7]A←A, (( $\rho T$ )÷( $-1 \uparrow T$ )*A)

[0]A←NO X
[1]AFN: TO NEGATE OUTCOME X
[2]ARA: LOGICAL INPUT
[3]ARE: NEGATION OF X
[4]A
[5]S← $\rho \rightarrow X[2]$ 
[6]X[2]←c $\rightarrow$ 'N','O','T',' ', $\rightarrow X[2]$ 
[7]A←X

[0]NONSTATMOD
[1]'NOT YET IMPLEMENTED'

[ 0]RES←MSDEV NORMCDF X;T;R
[ 1]A FN: calculate cumulative function values of a normal distribution
[ 2]A SS: iso/bs
[ 3]A RA: scalar or vector of normal statistics
[ 4]A LA: optional vector of length 2, mean and standard
[ 5]A LA: deviation of the distribution; default is standard dist. (0,1)
[ 6]A RE: scalar or vector, of normal cumulative distribution function
[ 7]A RE: values, of same length as right argument
[ 8]A AL: ?
[ 9]A
[10]A  $\square$ ELX←' $\square$ ERROR 'ARGUMENT ERROR''
[11] $\rightarrow$ (0= $\square$ NC 'MSDEV') $\uparrow$ con
[12]X←(X-MSDEV[1])÷MSDEV[2]
[13]con:
[14]R←(( $\rho X$ ) $\rho$ 0) $\circ$ .+(*-T $\times$ T÷9)÷T) $\times$ 1 $\circ$ (X $\circ$ . $\times$ (2*0.5) $\times$ T $\leftarrow$ 12)÷3
[15]RES←0 $\uparrow$ 1 $\downarrow$ 0.5+((X÷3 $\times$ 2*0.5) $\rightarrow$ +/ $\rho$ R)÷01

```

```

[ 0]RES←MSDEV NORMDEN X
[ 1]A FN: returns values of the normal probability density function
[ 2]A SS: iso/bs
[ 3]A RA: scalar or vector of normal statistic
[ 4]A RA: values at which probabilities are required
[ 5]A LA: optional vector of length 2, mean and standard
[ 6]A LA: deviation of the distribution; default is standard dist. (0,1)
[ 7]A RE: vector of probabilities of same length as right argument
[ 8]A AL: ?
[ 9]A
[10]A ELX←'ERROR ' 'ARGUMENT ERROR''
[11]→(0=NC 'MSDEV')↑con
[12]X←(X-MSDEV[1])÷MSDEV[2]
[13]con:
[14]RES←(÷(02)*0.5)×*~0.5×X*2
[ 0]A←NORMEST X

```

```

[ 1]A FN: THIS FUNCTION ESTIMATES THE NORMAL DISTRIBUTION PARAMETERS.
[ 2]A LA: NOT AVAILABLE
[ 3]A RA: A VECTOR OF POSITIVE VALUES.
[ 4]A RS: ESTIMATES OF PARAMETERS OF THE NORMAL DISTRIBUTION.
[ 5]A SS:
[ 6]A AL: ALGORITHM----USING THE MEAN m, THE VARIANCE (MAXIMUM LIKELIHOOD)
s,
[ 7]A AND THE UNADJUSTED VARIANCE su.
[ 8]m←+/X÷ρX
[ 9]s←((+/X*2)~(ρX)×m*2)÷~1+ρX
[10]su←((-1+ρX)×s)÷ρX
[11]A←s,m
[ 0]A←s_u NORMLIKE X

```

```

[ 1]R FN: FUNCTION TO CALCULATE THE LOG LIKELIHOOD
[ 2]R FN: OF THE NORMAL DISTRIBUTION
[ 3]R RA: LIFETIME DATA
[ 4]R LA: THE PARAMETERS FOR THE NORMAL DISTRIBUTION
[ 5]R RE: A SCALER VALUE, THE LIKELIHOOD
[ 6]R
[ 7]s←s_u[1]
[ 8]u←s_u[2]
[ 9]R LP ← BOTTOM SHELF
[10]L←-((+/0.5*s*2)+((X-u)*2)÷2 s*2)+(+/s(1-LP((X-u)÷s))
[0]A←s_u NORMPDF X
[1]R FN : FUNCTION CALCUALTES THE PDF
[2]R FOR THE NORMAL DISTRIBUTION
[3]R RA : SET OF VALUES AT WHICH PDF IS REQUIRED A VECTOR
[4]R LA : VECTOR OF LENGTH 2 1ST ARGUMENT SD 2ND MEAN
[5]R RE: A SACLER, PDF FOR NORMAL
[6]R
[7]s←s_u[1]
[8]u←s_u[2]
[9]A←(÷s*(2×01)*0.5)**-((X-u)*2)÷2×s*2
[ 0]RES←MSDEV NORMQ1 P;Q;U;T;Z1;Z2;C
[ 1]R FN: calculate normal statistic values or quantiles
[ 2]R SS: iso/bs
[ 3]R RA: scalar or vector of probabilities for which
[ 4]R RA: normal statistic required
[ 5]R LA: Optional argument - 2 item vector mean and standard deviation
[ 6]R RE: vector of normal statistics of same length as right argument
[ 7]R AL: Bailey(1981),Applied Statistics P275
[ 8]R
[ 9]→(0≠INC 'MSDEV')†con
[10]MSDEV←0 1
[11]con:
[12]U←(C←1E-6)†Q←P[1-P
[13]U←T×T←(0†-0.5×4×U×(1-U))*0.5

```

```

[14] Z1←T×1+U×0.008+U×-60.0003+4.37E-6×U
[15] U←-2×C1 Q
[16] Z2←T+(0.163+0.596÷T)÷T×T+(U-0.2×U)*0.5
[17] RES←,MSDEV[1]+MSDEV[2]×(×(P-0.5))×((1-T)×Z1)+Z2×T+Q<C
[ 0] RES←MSDEV NORMQUANT1 P;Q;U;T;Z1;Z2;C
[ 1]A FN: calculate normal statistic values or quantiles
[ 2]A SS: iso/bs
[ 3]A RA: scalar or vector of probabilities for which
[ 4]A RA: normal statistic required
[ 5]A LA: Optional argument - 2 item vector mean and standard deviation
[ 6]A RE: vector of normal statistics of same length as right argument
[ 7]A AL: Bailey(1981),Applied Statistics P275
[ 8]A
[ 9]A  ELX←'ERROR ' 'ARGUMENT ERROR' '
[10]←(0≠NC 'MSDEV')†con
[11]MSDEV←0 1
[12]con:
[13]U←(C+1E-6)†Q+P11-P
[14]U←T×T+(0†-0.5×4×U×(1-U))*0.5
[15]Z1←T×1+U×0.008+U×-60.0003+4.37E-6×U
[16]U←-2×C1 Q
[17]Z2←T+(0.163+0.596÷T)÷T×T+(U-0.2×U)*0.5
[18] RES←,MSDEV[1]+MSDEV[2]×(×(P-0.5))×((1-T)×Z1)+Z2×T+Q<C
[ 0] RES←NORMQUANT2 P
[ 1]A FN: calculate normal statistic values or quantiles
[ 2]A SS: iso/bs
[ 3]A RA: scalar or vector of probabilities for which
[ 4]A RA: normal statistic required
[ 5]A LA: Optional argument - 2 item vector mean and standard deviation
[ 6]A RE: vector of normal statistics of same length as right argument
[ 7]A AL: 1 ITERATION OF NORMTAIL
[ 8]A
[ 9]A  ELX←'ERROR ' 'ARGUMENT ERROR' '
[10] RES←NORMQUANT1 P

```

```

[11]P+(2*P)-1-NORMTAIL1 RES
[12]RES+NORMQUANT1 P
  [ 0]RES+NORMQUANT3 P
[ 1]R FN: calculate normal statistic values or quantiles
[ 2]R SS: iso/bs
[ 3]R RA: scalar or vector of probabilities for which
[ 4]R RA: normal statistic required
[ 5]R LA: Optional argument - 2 item vector mean and standard deviation
[ 6]R RE: vector of normal statistics of same length as right argument
[ 7]R AL: 1 ITERATION OF NORMTAIL
[ 8]R
[ 9]RES+NORMQUANT1 P
[10]P+(2*P)-1-NORMTAIL1 RES
[11]RES+NORMQUANT1 P
  [ 0]RES+MSDEV NORMTAIL1 X;Z;T;Q1;Q2;L;R
[ 1]R FN: calculate upper tail areas of standard normal distribution
[ 2]R SS: iso/bs
[ 3]R RA: scalar or vector of normal statistics
[ 4]R LA: optional vector of length 2, mean and standard
[ 5]R LA: deviation of the distribution; default is standard dist. (0,1)
[ 6]R RE: vector, of proportion of area in right tail of distribution,
[ 7]R RE: of same length as right arg.
[ 8]R AL:
[ 9]R
[10]R  $\square$ ELX+ 'ERROR 'ARGUMENT ERROR''
[11]+(0= $\square$ NC 'MSDEV') $\dagger$ con
[12]X+(X-MSDEV[1]) $\div$ MSDEV[2]
[13]con:
[14]R+((( $\rho$ X) $\rho$ 0) $\circ$  .+(*-T $\times$ T $\div$ 9) $\div$ T) $\times$ 1 $\square$ (X $\circ$  . $\times$ (2*0.5) $\times$ T $\div$ 12) $\div$ 3
[15]RES+,1-0 $\square$ 1 $\square$ 0.5+((X $\div$ 3 $\times$ 2*0.5) $\div$ +/R) $\div$  $\square$ 1
  [ 0]RES+NORMTAIL2 X;A;Y;Z;RES1;RES2

[ 1]R FN: calculates the tail probabilities of normal distribution

```

```

[ 2]R FN: P(X>right argument)

d
[ 3]R RA: any shape - point to the right of which probabilities are calculated

[ 4]R RE: an array of tail probs same size as right hand arg.

[ 5]R AL: Algorithm AS66

[ 6]R

[ 7]R ELX←'ERROR 'ARGUMENT ERROR''

[ 8]Z←|X-Y+X×X÷2

[ 9]A←0.398942280444 0.399903438504 5.75885480458 29.9213557808 2.62433121
679 48.6959930692 5.92885724438

[10]RES1←0.5-Z×A[1]-A[2]×Y÷Y+A[3]-A[4]÷Y+A[5]+A[6]÷Y+A[7]

[11]A←0.398942280385 -3.8052E-8 1.00000615302 0.000398064794 1.98615381364
0.151679116635 5.29330324926 4.8385912808 15.1508972451 0.742380924027 30.789
933034 3.99019417011

[12]RES2←A[1]×(*-Y)÷Z+A[2]+A[3]÷Z+A[4]+A[5]÷Z+A[6]+A[7]÷Z+A[8]+A[9]÷Z+A[10]
+A[11] Z+A[12]

[13]RES←((Z≤1.28)×RES1)+(Z>1.28)×RES2

[14]RES←RES+(X<0)×1-2×RES

[0]A←NORMVALUE A
[1]A←1.96
[0]A←NOT B
[1]A←B
[2]→(B=-99)/0
[3]A←~A
[0]OBJECTIVES
[1]R THE AIM OF THIS FUNCTION IS TO DEFINE THE OBJECTIVES OF THE ANALYSIS

```



[2]R AND THE USER'S DESIRES

[3]'HAVE YOU ANY OBJECTIVES OR ARE YOU EXPLORING SYSTEM ? OBJ/EXPL'

[4]→('O'=1ρ□)/i

[5]EXPLORE

[6]i:TEST

[0]OLDAT

[1]'ENTER NAME OF DATA'

[2]N←□

[3]ΔMEN1[1;2]←c ' N,' (OLD DATA)'

[4]'DO YOU LIKE TO UPDATE ',N,' DATA ? YES/NO'

[5]→('YES'=3†Y←□)/c

[6]→p

[7]c:UPDATE2 X

[8]p:START

[ 0]OPNOTD

[ 1]R THE AIM OF THIS FUNCTION IS TO DEFINE A LIST OF OPTIONS AVAILABLE TO

[ 2]R THE USER TO CHOOSE FROM WHEN THERE IS NO TREND IN DATA

[ 3]'

[ 4]' THE FOLLOWING IS A SET OF OPTION AVAILABLE TO CHOOSE FROM: '

[ 5]'TO CHOOSE AN OPTION PLEASE TYPE THE COMMAND IN [ ] BRACKETS '

[ 6]'AND PRESS RETURN'

[ 7]'

[ 8]' 1 - TEST FOR INDEPENDENCE [TEST\_INDEPENDENCE (DATA NAME)]'

[ 9]' 2 - MAIN MENUE [START]'

[10]' 3 - ANY OTHER ACTIVITY THE [USER\_DESIRE]'

[11]' 4 - QUIT [END]'

[12]'

[13]' PLEASE ENTER YOUR CHOSEN OPTION '

[ 0]OPTD

[ 1]R THE AIM OF THIS FUNCTION IS TO DEFINE A LIST OF OPTIONS AVAILABLE TO

[ 2]R THE USER TO CHOOSE FROM WHEN THERE IS TREND IN DATA

[ 3]'

[ 4]'THE FOLLOWING IS A SET OF OPTIONS AVAILABLE TO CHOOSE FROM :'

```

[ 5]'TO CHOOSE AN OPTION PLEASE TYPE THE COMMAND IN [ ] BRACKETS '
[ 6]'AND PRESS RETURN'
[ 7]' '
[ 8]' 1 - NHPP                                     [NHPPFIT]'
[ 9]' 2 - OTHER NON - STATIONARY MODEL, e.g. CROWS MODEL [NONSTATMOD]'
[10]' 3 - MAIN MENUE                               [START]'
[11]' 4 - ANY OTHER ACTIVITY THE USER DESIRES     [USER_DESIRE]'
[12]' 5 - QUIT                                     [END]'
[13]' '
[14]' PLEASE ENTER YOUR CHOSEN OPTION '
[0]OPTDLR
[0]A←PARAMETERS C
[1]A←C[2]
[0]A←PARENTS C
[1]A←C[3]
[ 0]R←P PCTILE X;N;IN;TT;T;IO
[ 1]R FN: calculates percentiles of set of values
[ 2]R SS: iso/bs
[ 3]R RA: vector of values for which percentile is to be calculated
[ 4]R LA: vector of percentiles (all <100) to be calculated
[ 5]R RE: vector of percentiles
[ 6]R
[ 7]R ELX←'ERROR ' 'ARGUMENT ERROR'
[ 8]IO←1
[ 9]P←,P÷100
[10]X←X[4X]
[11]N←X
[12]R←0.5×+/X[1N]TT,[1.5]TT+(TT+T)=T×P×N]
[ 0]A←POISSONEST X
[ 1]RA←POISSONEST X
[ 2]R FN: THIS FUNCTION ESTIMATES THE POISSON DISTRIBUTION PARAMETERS.
[ 3]R LA: NOT AVAILABLE
[ 4]R RA: VECTOR OF POSITIVE VALUES.
[ 5]R RS: ESTIMATES OF PARAMETERS OF THE POSSION DISTRIBUTION.

```

```

[ 6]ASS:
[ 7]AAL: ALGORITHM---- USING THE SAMPLE MEAN m.
[ 8]m←+/X÷ρX
[ 9]A←m
[10]AA
[0]PRINTFUNCTION NAME;S
[1]S←' ','[',(⌊(X,1)ρ-1+⌊X+1↑ρNAME),']',((NAME←⌊CR NAME),⌊TC[2]),⌊TC[3]
[2]PRINT←' ',' ',' ',' ',' ',' ',' ',S
[0]PRINT_ALL;A
[1]A←⌊NL 3
[2]N←1↑ρA
[3]J←0
[4]L:PRINT←' '
[5]PRINT ' '
[6]PRINTFUNCTION A[J+J+1;]
[7]→(J<N)/L
[0]RES←RANK X;C;I;RO;RO2;O
[1]A FN: returns the ranks of a data set
[2]A SS: iso/bs
[3]A RA: vector of data values
[4]A RE: the ranks in ascending order of the data set
[5]A
[6]A ⌊ELX←'⌊ERROR ''ARGUMENT ERROR'''
[7]X←X[O←4X]
[8]RES←0.5×(44X)+44X
[9]RES[O]←RES
[ 0]REPORT_INDEPENDENC
[ 1]A THE AIM OF THIS FUNCTION IS TO INSERT IF DATA IS
[ 2]A DEPENDENT/INDEPENDENT
[ 3]REPIND1←'INDEPENDENCE YES'
[ 4]REPIND2←'INDEPENDENCE NO'
[ 5]→(ΔI=1)/i
[ 6]→(ΔI=0)/e
[ 7]e:REPIND←REPIND2

```

```

[ 8]→c
[ 9]i:REPIND←REPIND1
[10]c:REPORT[2;]←c REPIND
[0]REPORT_LAPLACE
[1]INITIAL_REPORT
[2]R THE AIM OF THIS FUNCTION IS TO REPORT THE RESULT OF LAPLACE TEST
[3]→(ΔA=1)/i
[4]→(ΔA=0)/n
[5]n:REPLAP←REPLAP2
[6]→s
[7]i:REPLAP←REPLAP1
[8]s:REPORT[1;]←c REPLAP
[0]A←RESET
[1]A←1 1p ' '
[ 0]R_INDEP
[ 1]R THE AIM OF THIS FUNCTION IS TO INSERT IF DATA IS
[ 2]R DEPENDENT/INDEPENDENT
[ 3]REPIND1←'INDEPENDENCE' YES'
[ 4]REPIND2←'INDEPENDENCE' NO'
[ 5]→(ΔI=1)/i
[ 6]→(ΔI=0)/e
[ 7]e:REPIND←REPIND2
[ 8]→c
[ 9]i:REPIND←REPIND1
[10]c:REPORT[2;]←c REPIND
[ 0]A←R_LPTRD A
[ 1]R FN: REPORTS FROM THE LAPLACE TREND TEST
[ 2]R RA: SCALAR 0 (FAILURE) , 1 (SUCCESS) OR -99 (NO RESULT)
[ 3]R RE: SCALAR 0 (FAILURE) , 1 (SUCCESS) OR -99 (NO RESULT)
[ 4]R
[ 5]→(A≠0)/L1
[ 6]V←c (c 'NO TREND - LP') (c ΔLPTRD)
[ 7]→K
[ 8]L1:→(A=-99)/L2

```

```

[ 9]V←c (c 'TREND - LP')(cΔLPTRD)
[10]→K
[11]L2:V←c (c 'CHECK DATA')
[12]K:REPORT REPORT UPDATE_REP V
[ 0]A←R_MILTRD X
[ 1]A FN: REPORTS FROM THE MIL-HDBK-189 TREND TEST
[ 2]A RA: SCALAR 0 (FAILURE) OR 1 (SUCCESS) OR -99 (CHECK DATA)
[ 3]A RE: SCALAR 0 (FAILURE) OR 1 (SUCCESS)
[ 4]A
[ 5]→(X≠0)/L1
[ 6]V←c (c 'NO TREND - MIL')(cΔMILTRD)
[ 7]→K
[ 8]L1:→(X=-99)/L2
[ 9]V←c (c 'TREND - MIL')(cΔMILTRD)
[10]→K
[11]L2:V←c (c 'CHECK DATA')
[12]K:REPORT←REPORT UPDATE_REP V
[13]A←X
[0]RES←SDEV ARRAY;RO
[1]A FN: calculates standard deviation across first dimension
[2]A FN: of a set of values
[3]A SS: iso/bs
[4]A RA: array of values for which standard deviation is to be calculated
[5]A RE: standard deviation, dimensions are those of
[6]A RE: right argument having dropped the first element
[7]A
[8]RES←((+/(ARRAY-RO, (+/ARRAY)÷1+RO)*2)÷-1+1+RO, ARRAY÷1/ARRAY)*0.5
[0]RES←SDEVFT FTAB;RO
[1]A FN: calculates standard deviation of a frequency table
[2]A FN: (divisor n-1)
[3]A SS: iso/bs
[4]A RA: matrix where first column is X value and
[5]A RA: second column is frequency
[6]A RE: scalar standard deviation

```

```

[7]A
[8]A  ELX←'ERROR ' 'ARGUMENT ERROR' ''
[9]RES←((+/FTAB[;2]×(FTAB[;1]-(+/×/FTAB)÷RO)*2)÷-1+RO+(+FTAB)[2])*0.5
[ 0]RES←SKEWNESS ARRAY;M;R;RO;RO2
[ 1]A FN: calculates skewness of set of values
[ 2]A SS: iso/bs
[ 3]A RA: array of values for which skewness is to be calculated
[ 4]A RE: skewness, dimensions are those of right argument minus first.
[ 5]A
[ 6]A  ELX←'ERROR ' 'ARGUMENT ERROR' ''
[ 7]M←ARRAY-RO÷(+ARRAY)÷RO2+1+RO÷ARRAY+1/ARRAY
[ 8]R←(+M*3)÷RO2-1
[ 9]RES←R÷((+M*2)÷RO2-1)*1.5
[10]
[0]RES←DATA1 SPEARMAN DATA2;RO
[1]A FN: calculate Spearmans correlation coefficient
[2]A SS: iso/bs
[3]A RA: data vector
[4]A LA: data vector of same length as right argument
[5]A RE: Spearmans correlation coefficient
[6]A
[7]A  ELX←'ERROR ' 'ARGUMENT ERROR' ''
[8]RES←1-(6×+/(DATA1-DATA2)*2)÷(RO*3)-RO÷DATA2
[9]
[0]A←START
[1]△MENU1
[2]A←COLLECT '12345'
[3]'HELLO'
[4]⊞←'START',A
[0]START1
[1]DATA
[2]MENU
[0]START2
[1]OBJECTIVES

```

```

[2]MENU
  [0]START3
[1]MODELS
[2]MENU
  [0]START4
[1]KNOWLEDGE
[2]MENU
  [0]START5
[1]END
  [ 0]STATS DATA;T;U;R2;R;RO;RO2;V;W
[ 1]R FN: returns calculates the statistics of a set of values
[ 2]R FN: - min, 25 pctl, median, 75 pctl,
[ 3]R FN: max, mean, sdev, skew, kurtosis, semiqr, range
[ 4]R SS: iso/bs
[ 5]R RA: vector of data values
[ 6]R
[ 7]R DELX←'ERROR ' 'ARGUMENT ERROR' '
[ 8]'MIN      ',R[DATA-1]/DATA
[ 9]T←T-R+T*0.5+0.25*RO+DATA[DATA[DATA]]
[10]'25 PCTILE ',R+0.25 QUANTILE DATA
[11]'MEDIAN    ',R+0.5 QUANTILE DATA
[12]'75 PCTILE ',R+0.75 QUANTILE DATA
[13]'MAX      ',R[DATA]
[14]'MEAN      ',R+(T+DATA)/RO2+1+RO
[15]'VAR       ',R+U+((DATA-RO+T/RO2)*2)/-1+RO2
[16]'SDEV      ',R+U*0.5
[17]T←DATA-RO+T/RO2
[18]R←(T+T*3)/RO2-1
[19]'SKEW      ',R+R/(R2+(T+T*2)/RO2-1)*1.5
[20]R←(T+T*4)/RO2-1
[21]'KURT      ',R+R/R2*2
[22]'SEMIQR    ',R+(W-V)/2
[23]'RANGE     ',R+(R[DATA]-L[DATA])
[0]A←STNORM LEV

```

```

[1]A←1.96
[0]SYSTEM
[1]R THE AIM OF THIS FUNCTION IS TO START THE SYSTEM.
[2]INITIALMENU
[3]INITIALMEN2
[4]MENU
[0]A←K S_INDEP X
[1]R
[2]A←(K+X)CORR(-K)+X
[0]A←S_MILTRD X
[1]R FN: CALCULATES MIL-HDBK-189 TREND TEST STATISTIC
[2]R RA: VECTOR OF POSITIVE INCREASING NUMBERS
[3]R RE: SCALAR TEST STATISTIC
[4]R
[5]A←2x+/(1+X)÷-1+X
[0]RES←DEG TCDF T;TH;B;V;M;N;C
[1]R FN: calculate cumulative function of student t distribution
[2]R SS: iso/bs
[3]R RA: scalar or vector of t values
[4]R LA: scalar degree of freedom
[5]R RE: vector of t cumulative distribution function values
[6]R RE: of same length as right argument
[7]R AL: ?
[8]R
[9]R ELX←'ERROR ' 'ARGUMENT ERROR' '
[10]TH←-30|,T÷DEG*0.5
[11]B←(DEG=1)+(2×0=2|DEG)+3×(0≠2|DEG)^DEG÷1
[12]→(B=13)/one,even,odd
[13]one:RES←0.5+(xT)×TH÷01
[14]→0
[15]even:V←1,(M÷01)/÷\1M←-2+2×N←L DEG÷2
[16]C←(20TH)0.*0,2×1N-1
[17]RES←0.5+(xT)×0.5×(10TH)×C+.xV
[18]→0

```



```

[19]odd:V+1÷(M+1 0)/÷\M+1+2×N÷[DEG÷2
[20]C+(20TH)0.*-1+2×N
[21]RES+0.5+(XT)×(÷01)×TH+(10TH)×C+.×V
[ 0]RES+DF TDEN X;Y;Z
[ 1]R FN: returns values of the student t probability density function
[ 2]R SS: iso/bs
[ 3]R RA: scalar or vector of values at which probabilities are required
[ 4]R LA: number of degrees of freedom of student t distribution
[ 5]R RE: vector of probabilities of same length as right argument
[ 6]R AL: ?
[ 7]R
[ 8]R ELX+ 'ERROR ' 'ARGUMENT ERROR' '
[ 9]Y+(X/!Z-1)÷!+/-1,Z+0.5×1,DF
[10]RES+,÷(DF*0.5)×Y×(1+(X*2)÷DF)*+/Z
[0]A+TDHM T;X
[1]X+T-0,-1+T
[2]A+TRENDHM X
[0]A+TDLR T
[1]U+TDHM T
[2]CVX+CV T-0,-1+T
[3]UX+U÷CVX
[4]A+UX
[0]A+TDMILHDBK X
[1]R RA: CUMULATIVE FAILURE TIMES, TIMES AT WHICH FAILURES TAKE PLACE.
[2]A+-2×+/( (-1+X)÷(-1+X))
[ 0]TESTS
[ 1]'PLEASE ENTER YOUR CHOICE NUMBER'
[ 2]Δg+
[ 3]÷(1+Δg=1)/i
[ 4]÷(1+Δg=2)/j
[ 5]÷(1+Δg=3)/q
[ 6]÷(1+Δg=4)/k
[ 7]÷(1+Δg=5)/l
[ 8]÷(1+Δg=6)/n

```

```

[ 9]i:TEST_LAPLACE X
[10]MEN2b[1;]←c '          TESTED'
[11]→m
[12]j:TEST_INDEPENDENCE X
[13]MEN2b[2;]←c '          TESTED'
[14]→m
[15]q:TEST_PDF X
[16]MEN2b[3;]←c '          TESTED'
[17]→m
[18]k:TEST_LIKELIHOOD X
[19]MEN2b[4;]←c '          TESTED'
[20]→m
[21]l:TEST_LIFE X
[22]MEN2b[5;]←c '          TESTED'
[23]m:MENU
[24]→0
[25]n:END

[0]A←TEST_GOODFIT X
[1]R FN:
[2]R LA:
[3]R RA:
[4]R AL:
[5]R
[6]A←CHECK_GOODFIT X
[7](((FUNCTIONS)))
[8]A←COMPARE_GOODFIT X
[9]A←REPORT_GOODFIT X
[ 0]A←TEST_LAPLACE X
[ 1]R FN: DRIVE LAPACE TEST
[ 2]R LA: THERE IS NO LEFT ARGUMENT
[ 3]R RA: DATA AS A SERIES OF FAILURE TIMES
[ 4]R AL: ACHER AND FEINGOLD + RELIAB REPARABLE SYSTEMS PAGE NO
[ 5]R
[ 6]A←CHECK_LAPLACE X

```

```

[ 7]→(A=0)/0
[ 8]A←TRENDHM X
[ 9]A←COMPARE_LAPLACE X
[10]REPORT_LAPLACE
[0]A←TEST_LIFETIME X
[1]R FN:
[2]R LA:
[3]R RA:
[4]R AL:
[5]R
[6]A←CHECK_LIFETIME X
[7](((FUNCTIONS)))
[8]A←COMPARE_LIFETIME X
[9]A←REPORT_LIFETIME X
[0]A←TEST_LIKELIHOOD X
[1]R FN:
[2]R LA:
[3]R RA:
[4]R AL:
[5]R
[6]A←CHECK_LIKELIHOOD X
[7]A←(((FUNCTIONS)))
[8]A←COMPARE_LIKELIHOOD X
[9]A←REPORT_LIKELIHOOD X
[0]A←TEST_PDF X
[1]R FN:
[2]R LA:
[3]R RA:
[4]R AL:
[5]R
[6]CHECK_PDF X
[7]R (((FUNCTIONS)))
[8]COMP_PDF X
[9]REPORT_PDF X

```

```

[0]TIDYRDADIST
[1]NAMES←NL 3
[2]'[0] TIDYRDADIST
[3]','(1+P(NAMES),1)P(1+P(NAMES))',' ','E','X',' ',NAMES
[ 0]RES←DF TQUANT P;R;T
[ 1]R FN: calculate t statistic values or quantiles
[ 2]R SS: iso/bs
[ 3]R RA: scalar or vector of probabilities for which
[ 4]R RA: t statistic required
[ 5]R LA: scalar degree of freedom
[ 6]R RE: vector of t statistics of same length as right argument
[ 7]R AL: ?
[ 8]R
[ 9]R ELX←'ERROR ' 'ARGUMENT ERROR' '
[10]→(0=T×P-0.5)†(RES←0)
[11]P←2×(P,1-P)[1+P>0.5]
[12]→((DF=2),DF>2)/con1,con2
[13]RES←T×3÷0.5×P
[14]→0
[15]con1:
[16]RES←T×(2×1÷P×2-P)*0.5
[17]→0
[18]con2:
[19]R←0.076×(P×(-0.5+(0.2)*0.5)×DF*0.5)*DF
[20]RES←T×R+0.0953+(-0.631÷1+DF)+0.81×(-P×2-P)*0.5
[0]RES←K TQUANT2 P;W
[1]R FN: Produces t quantiles for given values
[2]R RA: An array of probabilities for which you require the quantiles
[3]R LA: degree of freedom for the t
[4]R RE: An array of quantiles same size as right hand argument
[5]R AL: Approximation BSC P 85
[6]R
[7]R ELX←'ERROR ' 'ARGUMENT ERROR' '
[8]W←(NORMQUANT1 P)×1+2÷1+8×K

```

```

[9]RES←(X P-0.5)×(K×-1+*W×W÷K)*÷2
[ 0]RES←K TQUANT3 X;X1;X2;RES2;ERR
[ 1]R FN: calculate t statistic values or quantiles
[ 2]R SS: iso/bs
[ 3]R RA: scalar or vector of probabilities for which
[ 4]R RA: t statistic required
[ 5]R LA: scalar degree of freedom
[ 6]R RE: vector of t statistics of same length as right argument
[ 7]R AL: Iteration of TTAIL1
[ 8]R
[ 9]R ELX←'ERROR ' 'ARGUMENT ERROR''
[10]ERR←1E-6
[11]RES←K TQUANT2 X
[12]loop:X1←1-K TTAIL1 RES
[13]→(ERR≥|X1-X)/0
[14]RES2←RES×X÷X1
[15]X2←1-K TTAIL1 RES2
[16]→((K=1)^X>0.993)/df1
[17]RES←RES-(RES-RES2)×(X-X1)÷X2-X1↪loop
[18]df1:RES←RES-(RES-RES2)×100×(1-X)×(X-X1)÷X2-X1↪loop
[ 0]TRANSFER C
[ 1]WE TRANSFER C
[ 2]R CREATE THE FILE DOR44
[ 3]('A:DOR',(X-WE),'.EXT')RNC CREATE WE
[ 4]DMAT←24 80 PAGE
[ 5]1←1
[ 6]L:→(1>1÷DMAT)/E
[ 7]NAME←DMAT[1;]
[ 8],NAME
[ 9]NAME RNC APPEND WE
[10]1←1+1
[11]→L
[12]E:NUNTIE WE
[0]A←TRENDHM X

```

```

[1]A←(((+/-1+X)÷(ρX)-1)-((-1+X)÷2))÷((-1+X)×(÷(12×((ρX)-1)))×0.5)
[0]A←TRENDLR X
[1]R X←T-0,-1+T
[2]EX←((ρX)×((ρX)-1))÷4
[3]VRX←(2×(ρX*3)+3×(ρX*2)-5ρX)÷72
[4]CVX←(VRX*0.5)÷EX
[5]U←CV X
[6]UX←U÷CVX
[7]EX,VRX,CVX,UX
[ 0]RES←MV TSTAT X;VARX;MX;RO
[ 1]R FN: calculates the one-sample t statistic for testing
[ 2]R FN: the hypothesis of the mean value of a dataset
[ 3]R SS: iso/bs
[ 4]R RA: vector of values of dataset
[ 5]R LA: optional - scalar, mean value to test (default 0)
[ 6]R RE: scalar t statistic
[ 7]R
[ 8]R □ELX←'□ERROR ''ARGUMENT ERROR'''
[ 9]←(0≠□NC 'MV')†con
[10]R test for mean, 0 if not given
[11]MV←0
[12]con:
[13]VARX←(+/(X-MX←(+/X)÷RO)*2)÷-1+RO←ρX
[14]RES←(MX-MV)÷(VARX÷RO)*0.5
[ 0]RES←DF TTAIL1 T;A;B;IOE;S;N;SER

[ 1]R FN: caculates the tail probabilities for Student's t distribution
[ 2]R FN: P(X>right argument)
[ 3]R FN: Algorithm AS3
[ 4]R RA: any shape - points which probabilities to right are calculated
[ 5]R LA: number of degrees of freedom (a scalar -common to all RA values)
[ 6]R RE: an array of t tail probs of same size as right hand argument.
[ 7]R
[ 8]R □ELX←'□ERROR ''ARGUMENT ERROR'''

```

```

[ 9]B←DF÷DF+T*2
[10]A←T÷DF*2
[11]IOE←2|DF
[12]S←1-DF=1
[13]→(DF<4)/1t4
[14]N←1+(DF-4+IOE)÷2
[15]SER←IOE+2×\N
[16]S←1++/×\B×.×(-1+SER)÷SER
[17]1t4:RES←(IOE=0)×0.5-0.5×A×S×B*2
[18]RES←RES+(IOE=1)×0.5-((-3×A)+A×B×S)÷0.1
[ 0]RES←X TWOAMT Y;VARXY;SUMX;SUMY;MX;MY;ROY;ROX
[ 1]A FN: calculates the two-sample t statistic for testing
[ 2]A FN: the hypothesis of equal means of two datasets,
[ 3]A FN: assuming equal variances
[ 4]A SS: iso/bs
[ 5]A RA: vector of values of first dataset
[ 6]A LA: vector of values of second dataset
[ 7]A RE: scalar t statistic
[ 8]A
[ 9]A DELX←'ERROR ''ARGUMENT ERROR'''
[10]SUMX←(+/(X-MX←(+/X)÷ROX←ρX)*2)
[11]SUMY←(+/(Y-MY←(+/Y)÷ROY←ρY)*2)
[12]VARXY←(SUMY+SUMX)÷ROY+ROX-2
[13]RES←(MX-MY)÷(VARXY×(÷ROY)÷ROX)*0.5
[ 0]RES←TWOWAY OBS;NT;T;B;MP;MB;WMS;BT;BMS;NPOP;CF;PTOTS;DG1;DG2;N
[ 1]A FN: calculate twoway analysis of variance without replications
[ 2]A SS: iso/bs
[ 3]A RA: matrix of observations where each row representing a block
[ 4]A RA: and each column a treatment or population
[ 5]A RE: a 4 x 4 array with 1st column DF, 2nd column SS, 3rd column
[ 6]A RE: MS and 4th column F-Ratio's
[ 7]A
[ 8]A DELX←'ERROR ''ARGUMENT ERROR'''
[ 9]A no. of pops and pop totals

```

```

[10]NPOP←ρPTOTS←+/OBS
[11]atotal and obs per pop
[12]NT←NPOP×N+1+ρOBS
[13]apop and block means
[14]MP←PTOTS÷N
[15]MB←(+/OBS)÷NPOP
[16]CF←((+/OBS←,OBS)*2)÷NT
[17]atotal, block and between sum of squares
[18]T←(+/OBS*2)-CF
[19]B←(+/NPOP×MB*2)-CF
[20]BT←(+/N×MP*2)-CF
[21]abetween and within mean squares
[22]BMS←BT÷DG1←NPOP-1
[23]WMS←(T-BT+B)÷DG2←NT-DG1+N
[24]aF ratio and degrees of freedom
[25]RES←DG1,(N-1),DG2,(DG1+DG2+N-1),BT,B,(T-BT+B),T,BMS,(B÷N-1),WMS,0
[26]RES←4 4ρRES,(BMS÷WMS),(B÷WMS×N-1),0,0
[ 0]A←Y T_INDEP X
[ 1]aFN: TO TEST THE INDEPENDENCE OF TWO SERIES Y AND X
[ 2]aRA: A VECTOR OF NUMERIC VALUES
[ 3]aLA: A VECTOR OF NUMERIC VALUES SAME LENGTH AS RA
[ 4]aRE: 1 + DEPT : 0 NOT DEPT : -1 - DEPT : -99 NO RESULT
[ 5]a
[ 6]A←CH_INDEP X
[ 7]←(A=1)/L1
[ 8]A←-99
[ 9]←0
[10]L1:ΔCORR←Y CORR X
[11]A←CO_INDEP ΔCORR
[12]A←R_INDEP A
[ 0]A←T_LPTREND X
[ 1]aFN: TO CARRY OUT LAPLACE TREND TEST
[ 2]aRA: VECTOR OF VALUES REPRESENTING FAILURE TIMES
[ 3]aRA:  $T_i \leq T_{i+1}$ 

```



```

[ 4]ARE: 1 + TREND : -1 - TREND : 0 NO TREND : -99 NO RESULT
[ 5]A←CH_LPTRD X
[ 6]→(A=1)/L1
[ 7]A←-99
[ 8]→0
[ 9]L1:ΔLTRD←S_LPTRD X
[10]A←CO_LPTRD ΔLTRD
[11]A←R_LPTRD A
[0]A←N T_SERCORR DATA
[ 0]A←N T_SERINDEP X
[ 1]AFN: TO TEST THE NTH SERIAL CORRELATION OF SERIES TO SEE IF
[ 2]AFN: IS AUTOCORRELATED OR NOT
[ 3]ARA: A VECTOR OF NUMERIC VALUES
[ 4]ALA: A INTEGER SCALAR LESS THAN THE LENGTH OF RA
[ 5]ARE: 1 + SDEPT : 0 NO SDEPT : -1 - SDEPT : NO RESULT
[ 6]A
[ 7]A←Y CH_SINDEPT X
[ 8]A←-99
[ 9]→0
[10]L1:±'ΔSCORR', (±N), '←N AUTOCORR X'
[11]±'A←CO_SINDEP ΔSCORR', ±N
[12]A←R_INDEP A
[ 0]A←T_TRENDLP X
[ 1]AFN: TO CARRY OUT LAPLACE TREND TEST
[ 2]ARA: VECTOR OF VALUES REPRESENTING FAILURE TIMES
[ 3]ARA:  $T_i \leq T_{i+1}$ 
[ 4]ARE: 1 - TREND : 0 NO TREND : -99 NO RESULT
[ 5]A←CH_LPTRD X
[ 6]→(A=1)/L1
[ 7]'NO RESULT CHECK DATA'
[ 8]A←R_LPTRD A
[ 9]→0
[10]L1:ΔLPTRD←S_LPTRD X
[11]A←CO_LPTRD ΔLPTRD

```

```

[12]A←R_LPTRD A
[ 0]A←T_TRENDMIL X
[ 1]A FN: TO CARRY OUT MIL-HDBK-189 TREND TEST
[ 2]A RA: VECTOR OF VALUES REPRESENTING FAILURE TIMES
[ 3]A RA:  $T_i \leq T_{i+1}$ 
[ 4]A RE: 1 + TREND : -1 - TREND : 0 NO TREND : -99 NO RESULT
[ 5]A←CH_MILTRD X
[ 6]→(A=1)/L1
[ 7]A←-99
[ 8]→0
[ 9]L1:ΔMILTRD←S_MILTRD X
[10]A←(2×-1+ρX)CO_MILTRD ΔMILTRD
[11]A←R_MILTRD A
[0]RES←UNIQUE DATA;I
[1]A FN: returns the unique values in a data set
[2]A SS: iso/bs
[3]A RA: vector of data values
[4]A RE: vector of the unique values in ascending order
[5]A AL: purposely avoids the use of *.
[6]A □ELX←'□ERROR ' 'ARGUMENT ERROR''
[7]DATA←,DATA
[8]I←1,1↓DATA≠-1↓DATA←DATA[↓DATA]
[9]RES←I/DATA
[ 0]A←UPDATE C;Y
[ 1]i:' DO YOU WANT TO ADD A COLUMN OR A ROW TO YOUR DATA ?ROW/COL'
[ 2]→('ROW'=1↑□)/g
[ 3]'WHAT IS THE NAME OF THE NEW COLUMN YOU WANT TO ADD ?'
[ 4]R←□
[ 5]'DO YOU WANT TO ADD THE NEW COLUMN ',R,' BEGINING/MIDDLE/END '
[ 6]' OF YOUR DATA MATRIX ?'
[ 7]→('B'=Y←1↑□)/a
[ 8]→('M'=Y)/b
[ 9]ENDCOL C
[10]→h

```

```

[11]g:' WHAT IS THE NAME OF THE NEW ROW YOU WANT TO ADD ?'
[12]L←□
[13]' DO YOU WANT TO ADD THE ROW ',L,' AT THE BEGINING /MIDDLE /END '
[14]'OF YOUR MATRIX ?'
[15]→('BEGINING'=Y1+□)/d
[16]→('M'=Y)/e
[17]ENDROW C
[18]→h
[19]a:C+BIGCOL C
[20]→h
[21]b:C+COLMIDDAT C
[22]→h
[23]c:ENDCOL C
[24]→h
[25]d:BIGROW C
[26]→h
[27]e:ROWMIDDAT C
[28]f:ENDROW C
[29]→h:'DO YOU WANT TO ADD ANY MORE DATA ? Y OR N'
[30]→('Y'=1+□)/i
[31]A←C
[ 0]A←UPDATE2 X;P1;P2;P3;V1;V2;V3;aa;bb;cc;dd;ff
[ 1]A THE AIM OF THIS FUNCTION IS TO UPDATE OLD DATA.
[ 2]ff:' DO YOU WANT TO ADD, ERASE OR CHANGE VALUES ? ADD/ERASE/CHANGE '
[ 3]W←□
[ 4]→('ADD'=3+W)/aa
[ 5]→('ERA'=3+W)/bb
[ 6]→('CHA'=3+W)/cc
[ 7]aa:'THE VALUES WILL BE ADDED ONE AT A TIME, PLEASE ENTER'
[ 8]'THE POSITION OF THE FIRST VALUE TO ADD'
[ 9]P1←□
[10]' NOW ENTER THE VALUE'
[11]V1←□
[12]A←□+((P1-1)÷X),V1,((P1-1)÷X)

```

```

[13]→dd
[14]bb: 'ENTER THE POSITION OF THE VALUE YOU WANT ERASED'
[15]P2←□
[16]A←□←(P2-1)ρX),((P2-1)↓X)
[17]→dd
[18]cc: ' ENTER THE POSITION OF THE VALUE YOU WANT TO CHANGE'
[19]P3←□
[20]' NOW ENTER THE NEW VALUE'
[21]V3←□
[22]A←□←((P3-1)ρX), (P3↓X)
[23]dd: 'DO YOU HAVE ANY MORE CHANGES TO DATA ? Y/N'
[24]→('Y'=1↑R←□)/ee
[25]MENUE2
[26]ee:→ff
    [0]C←B UPDATE_REP A
[1]R FN: TO UPDATE ARRAY B WITH A
[2]R RA: CURRENT FUNCTIONS REPORT AN SCALAR POSSIBLY NESTED
[3]R LA: AN ARRAY WITH N ROWS AND 1 COLUMN
[4]R RE: UPDATED ARRAY
[5]C←B,[1]A
    [0]USER_DESIRE
[1]R THE AIM OF THIS FUNCTION IS TO ENABLE THE USER DECIDE
[2]R THE NEXT STEP OF THE ANALYSIS PROCESS
[3]' PLEASE ENTER THE NUMBER OF YOUR CHOICE'
    [0]A←VAR X
[1]A←(+/(X-+/X÷ρX)*2)÷-1+ρX
    [ 0]A←a_b WEIBCDF X
[ 1]R FN: FUNCTION TO CALCULATE THE CUMULATIVE DISTRIBUTION FUNCTION
[ 2]R FN: FOR THE WEIBULL DISTRIBUTION.
[ 3]R SS:
[ 4]R RA: A VECTOR OF LIFETIMES
[ 5]R LA: a = WEIBULL SCALE PARAMETER, b = WEIBULL SHAPE PARAMETER
[ 6]R RE: CUMULATIVE DISTRIBUTION FUNCTION FOR THE WEIBULL DISTRIBUTION
[ 7]R

```

```

[ 8]A
[ 9]a←a_b[1]
[10]b←a_b[2]
[11]A→(X≤0)/d
[12]A←1-*(X÷a)*b
[13]→0
[14]d:→0
    [0]A←WEIBEST2 X;LX;B;N;BO;e;U
[ 1]N←X
[ 2]LX←X
[ 3]B←1
[ 4]e←BO+B
[ 5]B←N÷+/LX×(U×N÷+/U×X*B)-1
[ 6]→(0.00001<|(B-BO))/e
[ 7]A←(N÷+/X*B),B
    [0]A←WEIBLIKE X
[ 1]A FN:
[ 2]A FN:
[ 3]A RA:
[ 4]A LA:
[ 5]A RE:
[ 6]A
    [0]A←a_b WEIBPDF X
[ 1]a←a_b[1]
[ 2]b←a_b[2]
[ 3]A←a×b×((a×X)*b-1)×*-(a×X)*b
    [0]A←a_b WIEBPDF t
[ 1]a←a_b[1]
[ 2]b←a_b[2]
[ 3]A←a×b×((a×t)*b-1)×*-(a×t)*b
    [0]RES←DATA1 WILCOXON DATA2;Z;Q
[ 1]A FN: Wilcoxon test statistics (allows for rank ties)
[ 2]A SS: iso/bs
[ 3]A RA: data vector, representing 1st measurement

```

```

[4]R LA: data vector of same length as right argument, representing
[5]R LA: 2nd measurement
[6]R RE: scalar Wilcoxon test statistics
[7]R
[8]R DELX←'ERROR ''ARGUMENT ERROR'''
[9]RES←L/(+/0[Z],+/-0[Z*(XZ)×0.5×(A+Q)+A+Q|Z(0≠Z)/Z(DATA1-DATA2)
  [0]RES←DATA1 WILCOXONTS DATA2;T;Z;I;C;RANKS;VAR;RO
[1]R FN: Wilcoxon test statistic, Normalised form
[2]R SS: iso/bs
[3]R RA: data vector, representing 1st measurement
[4]R LA: data vector of same length as right argument, representing
[5]R LA: 2nd measurement
[6]R RE: scalar Wilcoxon test statistic, normalised form
[7]R
[8]R DELX←'ERROR ''ARGUMENT ERROR'''
[9]T←L/(+/0[Z],+/-0[Z*(XZ)×RANKS+0.5×(A+C)+A+C|Z(0≠Z)/Z(DATA1-DATA2)
[10]I←1,1+RANKS≠1+RANKS←RANKS[A+RANKS]
[11]C←((1+C),1+I)-C+I/√I
[12]VAR←((X/RO,1+1 2×RO+DATA1)-0.5×(-1+C*2)×C+/(1≠C)/C)÷24
[13]RES←(T-0.25×RO×RO+1)÷VAR*0.5
[0]ΔHOWΔCHICDF1
[1]ΔPRINT* '      3 CHICDF1 8'
[2]ΔPRINT*3 CHICDF1 8
[3]ΔPRINT* ' '
[4]ΔPRINT* '      1 CHICDF1 7'
[5]ΔPRINT*1 CHICDF1 7
[0]ΔHOWΔCHICDF2
[1]ΔPRINT* '      3 CHICDF2 8'
[2]ΔPRINT*3 CHICDF1 8
[3]ΔPRINT* ' '
[4]ΔPRINT* '      1 CHICDF2 7'
[5]ΔPRINT*1 CHICDF2 7
[0]ΔHOWΔCHIDEN
[1]ΔPRINT* '      1 CHIDEN 100'

```

```

[2]ΔPRINT*1 CHIDEN 100
[3]ΔPRINT* ' '
[4]ΔPRINT* '      4 CHIDEN VECTOR'
[5]ΔPRINT*4 CHIDEN VECTOR
[0]ΔHOWΔCHIGOF1
[1]ΔPRINT* '      VECTOR CHIGOF1 (VECTOR1)×(+/VECTOR)÷+/VECTOR1'
[2]ΔPRINT*VECTOR CHIGOF1(VECTOR1)×(+/VECTOR)÷+/VECTOR1
[0]ΔHOWΔCHIGOF2
[1]ΔPRINT* '      VECTOR CHIGOF1 (VECTOR1)×(+/VECTOR)÷+/VECTOR1'
[2]ΔPRINT*VECTOR CHIGOF2(VECTOR1)×(+/VECTOR)÷+/VECTOR1
[0]ΔHOWΔCHIQUANT1
[1]ΔPRINT* '      4 CHIQUANT1 .89'
[2]ΔPRINT*4 CHIQUANT1 0.89
[3]ΔPRINT* ' '
[4]ΔPRINT* '      5 CHIQUANT1 PVECTOR'
[5]ΔPRINT*5 CHIQUANT1 PVECTOR
[0]ΔHOWΔCHIQUANT2
[1]ΔPRINT* '      4 CHIQUANT2 .78'
[2]ΔPRINT*4 CHIQUANT2 0.78
[3]ΔPRINT* ' '
[4]ΔPRINT* '      1 CHIQUANT2 .8'
[5]ΔPRINT*1 CHIQUANT2 0.8
[0]ΔHOWΔCHIQUANT3
[1]ΔPRINT* '      4 CHIQUANT3 .89'
[2]ΔPRINT*4 CHIQUANT3 0.89
[3]ΔPRINT* ' '
[4]ΔPRINT* '      5 CHIQUANT3 PVECTOR'
[5]ΔPRINT*5 CHIQUANT3 PVECTOR
[0]ΔHOWΔCHITAIL1
[1]ΔPRINT* '      1 CHITAIL1 6'
[2]ΔPRINT*1 CHITAIL1 6
[3]ΔPRINT* ' '
[4]ΔPRINT* '      5 CHITAIL1 ARRAY2'
[5]ΔPRINT*5 CHITAIL1 ARRAY2

```

```

[0]ΔHOWΔCHITAIL2
[1]ΔPRINT* '      1 CHITAIL2 6'
[2]ΔPRINT*1 CHITAIL1 6
[3]ΔPRINT* ' '
[0]ΔHOWΔFQUANT1
[1]ΔPRINT* '      4 3 FQUANT1 .99'
[2]ΔPRINT*4 3 FQUANT1 0.99
[3]ΔPRINT* ' '
[4]ΔPRINT* '      10 6 FQUANT1 .80'
[5]ΔPRINT*10 6 FQUANT1 0.8
[0]ΔHOWΔFQUANT2
[1]ΔPRINT* '      4 3 FQUANT2 .99'
[2]ΔPRINT*4 3 FQUANT2 0.99
[3]ΔPRINT* ' '
[4]ΔPRINT* '      10 6 FQUANT2 .80'
[5]ΔPRINT*10 6 FQUANT2 0.8
[0]ΔHOWΔFQUANT3
[1]ΔPRINT* '      4 3 FQUANT1 .99'
[2]ΔPRINT*4 3 FQUANT1 0.99
[3]ΔPRINT* ' '
[4]ΔPRINT* '      10 6 FQUANT1 .80'
[5]ΔPRINT*10 6 FQUANT1 0.8
[0]ΔHOWΔFTAIL1
[1]ΔPRINT* '      2 3 FTAIL1 8'
[2]ΔPRINT*2 3 FTAIL1 8
[3]ΔPRINT* ' '
[4]ΔPRINT* '      5 6 FTAIL1 VECTOR'
[5]ΔPRINT*5 6 FTAIL1 VECTOR
[0]ΔHOWΔFTAIL2
[1]ΔPRINT* '      2 3 FTAIL2 8'
[2]ΔPRINT*2 3 FTAIL2 8
[3]ΔPRINT* ' '
[0]ΔHOWΔNORMCDF
[1]ΔPRINT* '      1 2 NORMCDF 4'

```



```

[2]ΔPRINT$ 1 2 NORMCDF 4
[3]ΔPRINT$ ' '
[4]ΔPRINT$ ' NORMCDF 3'
[5]ΔPRINT$NORMCDF 3
[6]ΔPRINT$ ' '
[7]ΔPRINT$ ' 4 5 NORMCDF VECTOR'
[8]ΔPRINT$4 5 NORMCDF VECTOR
[0]ΔHOWΔNORMDEN
[1]ΔPRINT$ ' 1 2 NORMDEN 4'
[2]ΔPRINT$ 1 2 NORMDEN 4
[3]ΔPRINT$ ' '
[4]ΔPRINT$ ' NORMDEN 3'
[5]ΔPRINT$NORMDEN 3
[6]ΔPRINT$ ' '
[7]ΔPRINT$ ' 4 5 NORMDEN VECTOR'
[8]ΔPRINT$4 5 NORMDEN VECTOR
[0]ΔHOWΔNORMQUANT1
[1]ΔPRINT$ ' 1 2 NORMQUANT1 .4'
[2]ΔPRINT$ 1 2 NORMQUANT1 0.4
[3]ΔPRINT$ ' '
[4]ΔPRINT$ ' NORMQUANT1 .8'
[5]ΔPRINT$NORMQUANT1 0.8
[6]ΔPRINT$ ' '
[7]ΔPRINT$ ' 4 5 NORMQUANT1 PVECTOR'
[8]ΔPRINT$4 5 NORMQUANT1 PVECTOR
[0]ΔHOWΔNORMQUANT2
[1]ΔPRINT$ ' NORMQUANT2 .4'
[2]ΔPRINT$NORMQUANT2 0.4
[3]ΔPRINT$ ' '
[4]ΔPRINT$ ' NORMQUANT2 .8'
[5]ΔPRINT$NORMQUANT2 0.8
[6]ΔPRINT$ ' '
[7]ΔPRINT$ ' NORMQUANT2 PVECTOR'
[8]ΔPRINT$NORMQUANT2 PVECTOR

```

```

[0]ΔHOWΔNORMQUANT3
[1]ΔPRINT* '      NORMQUANT3 .4'
[2]ΔPRINT* NORMQUANT3 0.4
[3]ΔPRINT* ' '
[4]ΔPRINT* '      NORMQUANT3 .8'
[5]ΔPRINT* NORMQUANT3 0.8
[6]ΔPRINT* ' '
[7]ΔPRINT* '      NORMQUANT3 PVECTOR'
[8]ΔPRINT* NORMQUANT3 PVECTOR

[0]ΔHOWΔNORMTAIL1
[1]ΔPRINT* '      1 2 NORMTAIL1 4'
[2]ΔPRINT* 1 2 NORMTAIL1 4
[3]ΔPRINT* ' '
[4]ΔPRINT* '      4 5 NORMTAIL1 VECTOR'
[5]ΔPRINT* 4 5 NORMTAIL1 VECTOR

[0]ΔHOWΔTCDF
[1]ΔPRINT* '      1 TCDF 8'
[2]ΔPRINT* 1 TCDF 8
[3]ΔPRINT* ' '
[4]ΔPRINT* '      6 TCDF VECTOR'
[5]ΔPRINT* 6 TCDF VECTOR

[0]ΔHOWΔTDEN
[1]ΔPRINT* '      1 TDEN 8'
[2]ΔPRINT* 1 TDEN 8
[3]ΔPRINT* ' '
[4]ΔPRINT* '      6 TDEN VECTOR'
[5]ΔPRINT* 6 TDEN VECTOR

[0]ΔHOWΔTQUANT
[1]ΔPRINT* '      1 TQUANT .8'
[2]ΔPRINT* 1 TQUANT 0.8
[3]ΔPRINT* ' '
[4]ΔPRINT* '      6 TQUANT PVECTOR'
[5]ΔPRINT* 6 TQUANT PVECTOR

[0]ΔHOWΔTQUANT2

```

```

[1]ΔPRINT* '      1 TQUANT2 .8'
[2]ΔPRINT*1 TQUANT2 0.8
[3]ΔPRINT* ' '
[4]ΔPRINT* '      6 TQUANT2 PVECTOR'
[5]ΔPRINT*6 TQUANT2 PVECTOR
[0]ΔHOWΔTQUANT3
[1]ΔPRINT* '      1 TQUANT3 .8'
[2]ΔPRINT*1 TQUANT3 0.8
[3]ΔPRINT* ' '
[4]ΔPRINT* '      6 TQUANT3 PVECTOR'
[5]ΔPRINT*6 TQUANT3 PVECTOR
[0]ΔHOWΔTSTAT
[1]ΔPRINT* '      5 TSTAT VECTOR'
[2]ΔPRINT*5 TSTAT VECTOR
[3]ΔPRINT* ' '
[4]ΔPRINT* '      TSTAT VECTOR'
[5]ΔPRINT*TSTAT VECTOR
[0]ΔHOWΔTTAIL1
[1]ΔPRINT* '      1 TTAIL1 8'
[2]ΔPRINT*1 TTAIL1 8
[3]ΔPRINT* ' '
[4]ΔPRINT* '      6 TTAIL1 ARRAY2'
[5]ΔPRINT*6 TTAIL1 ARRAY2
[0]ΔHOWΔWILCOXON
[1]ΔPRINT* '      VECTOR1 WILCOXON VECTOR'
[2]ΔPRINT*VECTOR1 WILCOXON VECTOR
[0]ΔHOWΔWILCOXONTS
[1]ΔPRINT* '      VECTOR1 WILCOXONTS VECTOR'
[2]ΔPRINT*VECTOR1 WILCOXONTS VECTOR
[0]ΔMENU1
[1]ΔΔMENU1
[2]ΔMEN1

```

## **APPENDIX SIX**

### **PUBLISHED PAPERS**

# ARDA - EXPERT SYSTEM FOR RELIABILITY DATA ANALYSIS

By Dr Jake Ansell and Mulhim Al-Doori  
Department of Business Studies  
The University of Edinburgh  
William Robertson Building  
50 George Square  
Edinburgh  
EH8 9JY  
Fax No. UK 031-668-3053  
Tel No. UK 031-650-3806

## Abstract

The paper explores the design and implementation of ARDA, an Expert System to analyse Reliability Data. Initially the viability of the knowledge domain is explored. The philosophy of design of the system is discussed. Details of the implementation are described. There is discussion of extension of system to other statistical analyses and of using alternative inferential bases.

**KEYWORDS:** EXPERT SYSTEMS, RELIABILITY DATA ANALYSIS, OBJECT ORIENTATED PROGRAMMING.

## Introduction

Many application of Expert Systems have been considered in Statistics. A few of these have been successful and the failures have been fruitful in highlighting the possible problems in encoding statistical knowledge into software. The problems are usually the scope of the area and its tractability. Many areas are still too large for a workable software system. In other areas there are difficulties manipulating the knowledge.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

© 1993 ACM 0-89791-612-3/93/0008/0001...\$1.50

In statistics there are two main types of Expert System: those which use statistical reasoning and those which assist statistical analysis. ARDA is an Expert System to assist Reliability Data Analysis (RDA) and is a member of the second group. We are not aware of a previously Expert Systems in this area. This might suggest the area is ill defined or the reasoning process not easily implemented. In the first section we review the knowledge domain, Reliability Data Analysis (RDA).

Whilst ARDA was originally conceived as an OOP system, ambiguities arose which meant we found the such an approach too confining. Hence the system is based on the flexibility of APL with elements taken from OOP Design. The philosophy of design is discussed in the second section. The third section covers ARDA's implementation. Obviously it is not possible to outline all aspects of ARDA in detail but an overview is given. The final section of the paper is a discussion how ARDA could be extended.

## Reliability Data Analysis

For an Expert System to be viable the knowledge domain needs to be well defined and sufficiently limited. Reliability has a well developed literature: texts by Ascher and Feingold [1984], Cox and Lewis [1966], Cox and Oakes [1984], Lawless [1982] and Mann, Schafer and Singpurwalla [1974]; a number of journals such as 'Reliability Engineering and Safety System' and 'IEEE Transactions on Reliability'; and relevant articles appear in a number of other journals. A significant portion of this material is devoted to Reliability Data

Analysis. There are also a number of packages, S-Plus, SAS and others including AGSS, an APL based system, see Lewis (1993), which enable the user to analyse the data using specific techniques. These packages do not give advice on selecting or applying the techniques.

For the system to be viable the boundaries need to be defined. ARDA has been designed to fit an appropriate model to data. This includes diagnosis of the stochastic processes and selecting an underlying distribution. The types of question the system may deal with are: 'What is the underlying distribution?', 'Should the component be replaced?' and 'Is the component under test better than the one currently being used?'. An issue of concern in building Expert Systems for statistical analysis is the strategies involved in the analysis. Whilst the philosophy underpinning statistics has been developed through statistical inference little attention has been paid to the practical problems of how a full analysis should be carried out. Cox and Snell [1971] do suggest some general strategies, but these still seem technique based. Others, Nelder [1990], have suggested empirical approaches. These approaches, though ultimately rewarding, would require considerable effort.

Currently the tools to build a system are relatively simple. Of the systems developed to aid analysis two types emerge: Knowledge Enhancing systems, KENS, or Knowledge Enabling systems, GLIMPSE. A Knowledge Enhancing system is a computerised text supplying the user with the required information. Knowledge Enabling system not only supply such knowledge but can implement the acquired knowledge. For a specific technique the system may suggest steps in the analysis. In GLIMPSE, for Generalised Linear Models, the system will give advice on the next step, it does not though concern itself with the full analysis.

In ARDA the objective of the analysis is used to provide the strategy rather than just employing techniques. Suppose the objective is to decide whether to replace a component or not. The technique based approach would be to find the precise distribution whilst the objective lead approach would simply require to establish whether the distribution has an increasing failure rate or not.

### Philosophy of Design

ARDA was initially conceived as an OOP system. Such systems have already been built in APL, see Alfonseca [1990] and Frey [1992]. Both these examples have extended APL to facilitate OOP design in APL. They have demonstrated the power of APL in creating features such as Abstraction, Encapsulation, Inheritance and Polymorphism. They have not indicated whether OOP is too limiting for APL design.

Obviously some features of ARDA are ideally suited to OOP. ARDA has a natural granular design. The 'System' would consist of three objects: 'Objective', 'Model' and 'Data' which conform to OOP principles. Each object would have a data structure, usually represented as a nested vector array, on which procedures would operate. There would be a hierarchy of Objects. Polymorphism is an apt description of the need to have different algorithms to deal with different forms of data producing the same output.

Whilst OOP has many advantages, in ARDA there was a major ambiguity in the main role of analysis. In the analysis elements from each of object 'Objective', 'Model' and 'Data' are often required simultaneously. This could be tackled at 'System' level but the supposed advantage of granularity would be lost. Alternatively one could treat 'data' ambiguously as an object and a message. In analysis 'data' would be treated as a message to the object 'model' which would be analysed in light of the elements from the object 'objective'. The hierarchical position of 'model' and 'objective' would therefore also need to be determined. Considerable effort could be involved in resolving these issues without a real return in software development.

The compromise made was to use strands from OOP with standard APL usage. The 'non-authoritarian' approach of ARDA is an example of the use of APL. The phrase 'non-authoritarian' was coined by Nelder [1990] to describe the approach taken in GLIMPSE. Analysts were given the freedom to pursue their own analysis without undue restriction. An analyst may even be allowed to use techniques 'incorrectly'. In such circumstances obviously it is unlikely that advice/guidance can be given.

This 'non-authoritarian' approach is implemented by using two levels of function within the workspace. The high level function will provide the expertise of the system, while the low level function will provide the basic calculations. Obviously the users will have the freedom to choose the level of functions they use. The high level function will generally test inputs to ensure correct usage. Underpinning the use of the high level functions will be a map for the analysis, see Figure 1. The map again does not assume that an analyst will follow a specific route but that the analyst will develop paths through the map. The map links the high level functions. Further discussion of this will be provided in the implementation section.

and stochastic processes. There are a set of procedures which operate on these vectors. A particularly useful set of function for analysis are those which construct the distribution trees, see MacDonald and Richards [1987].

## Objectives

Currently the least developed object is 'objective'. So far only basic objectives have been included in the object, though it is hoped to extend this later. The object aims to define and clarify the objective, and hence specify the strategy. The function for clarification and manipulation of the objectives are also limited. As with the distributions in the object 'model' for each of the objectives there will be a vector. This will contain the set of conditions the objective to be satisfied. These conditions will be nodes on the analysis map, see Figure 1. The path through the map is therefore defined and strategy selected.

## Analysis

Movement through the analysis path will be governed by the objective selected. This provides a control mechanism in ARDA. It will however be affected by the results obtained at each node in the map which may reroute the analysis or end it.

Each node in the map will have associated with it one or more low and high level functions. The low level function will provide the basic calculation, usually calculating a test statistic. The high level function acts as a driver for the low level function, providing guidance and advice. The general form for the function is given in Figure 2.

Figure 2

Idealised format of high level function, Test

```
Test
Check_Test
Test_Statistic
Compare_Test
Report_Test
```

The high level function will check the preconditions for test (Check\_Test), whether the data exists and is in the correct form. If the preconditions are not met a report will be produced which includes details of the low level function. If the preconditions are met then the low level function will be actioned and an appropriate comparison made (Compare\_Test). Based on the result of the comparison using a Frequentist approach a report will be produced. This will provide a basic interpretation and advice/guidance on the next stage of the

analysis. The user may or may not take the advice, a 'non-authoritarian' approach.

## Discussion

ARDA, an APL Expert System for Reliability Data Analysis, has been described in the paper. ARDA has been built partially using OOP approach, however where this has led to ambiguity or over elaboration we have used standard APL.

ARDA provides a model which could be used for a wide variety of data analysis. The system is innovatory in the use of objectives to guide analysis through the analysis map. To extend the system to other statistical analyses would require the definition of the set of objectives and models, and to develop an analysis map. The general framework would be usable, though, it may be also necessary to modify the 'Data' object.

Although we have described ARDA as used for data analysis it may be used to enhance the users knowledge. The links between Objectives, Models and Data can be explored either to find what data is required to establish a given objective or which models are related to which objectives. Hence it can provide a 'what-if' analysis. Such inversion reflects APL's natural flexibility.

Whilst currently using a Frequentist with hypotheses being accepted or rejected, there is no obvious impediment to generalising the approach to take account of the probabilities of the hypotheses. Then the satisfaction of the objectives could be expressed in terms of probabilities. This could be extended to the implementation of a Bayesian analysis. These alternatives are under consideration at the present time.

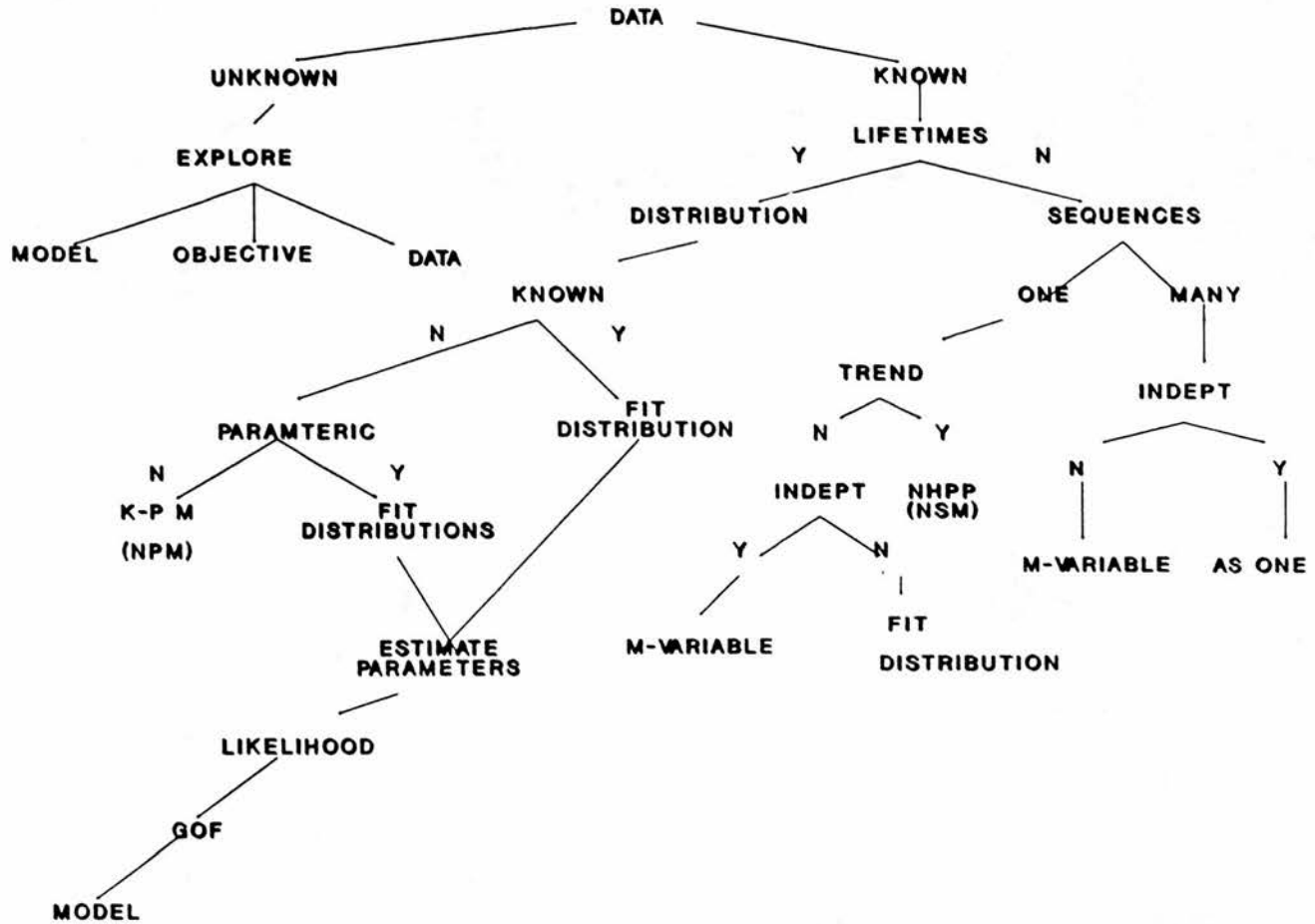
## References

- Alfonseca, M, [1990], 'Object Oriented Programming Tutorial', APL Quote Quad, 20, 7-8.
- Acher, H, and Fiengold, H, [1984], Reliability of Repairable Systems, Marcell Dekker: New York.
- Cox, DR, and Lewis, PAW, [1966], The Analysis of Series of Events, Methuen, London.
- Cox, DR, and Oakes, D, [1984], Analysis of Survival Data, Methuen, London.
- Cox, DR, and Snell, ES, [1971], 'Applied Statistics', Chapman Hall, London.
- Frey, RJ, [1992], 'Object Oriented Extensions to APL and J', Vector, 9, 116-125.



Figure 1

## The Analysis Map



### Implementation of the ARDA

It is not possible to describe ARDA in detail, but we will highlight the main features in this section.

The object based system uses nested arrays as the main data structure. The 'System' will consist of five elements (slots):

'Data', 'Model', 'Objective', 'Control' and 'Knowledge'. The first three will have vector nested arrays and are discussed later. 'Knowledge' and 'Control' will consist of procedures based on the 'Status Vector' of 'System'. 'Control' will generally consist of functions checking the 'Status Vector' to see if procedures can be carried out. 'Knowledge' at the 'System' level will consist of 'HOW' functions. These will indicate how a specific function operates, what its inputs and outputs are, and the detail of the algorithm involved.

### Data

This object is concerned with the entry and manipulation of data. The vector 'Data' will consist of the elements (slots):

Status, Definition, Validity, Structure and X, the actual data. The first four elements describe the data and its attributes, for example 'Status' indicates whether the data is defined or not. Associated with each element there will be a set of procedures, functions. INITIALD for example will create the array 'data' and set 'Status' to not defined.

### Model

This object is more complex than 'Data', dealing with the selection/defining of an appropriate model. It also contains information about the models. There are a number of vector nested arrays in the object. The vector 'Model' contains details of the current model with elements: Status, Stochastic and Distribution. 'Stochastic' describes the stochastic processes concerned and 'Distribution' names the underlying distribution. The other vectors contain knowledge on the distributions



IBM Research, (1992), A Graphical Statistical System (AGSS) Reliability Analysis, IBM Research, Yorktown Heights, New York.

Lawless, JF, [1982], Statistical Models and Methods for Lifetime Data, J Wiley & Sons, New York.

Lewis, PAW, [1993], 'A Graphical Statistical System AGSS', APL '93.

MacDonald, JB, and Richards, DO, [1987], 'Model Selection: Some generalized Distributions', Proc Reliability '87, Birmingham, UKAEA, Culcheth.

Mann, NR, Schafer, RE, and Singpurwalla, ND, [1974], Methods for the Statistical Analysis of Reliability and Life Data, J Wiley & Sons, New York.

Nelder, J, [1990], Talk given to RSS Seminar on Future of Statistical Computing', Royal Institution , London.

# THE RANDOM VECTOR

*The Newsletter of the APL Statistics Library*  
*Editor David Eastwood*

October 1992

## Contents

Editorial	David Eastwood	30
ASL - The Future	Jake Ansell	31
ASLGREG reviewed	Jon Sandles	32
Review comments	Alan Sykes	34
ASLGREG corrections	Alan Sykes	35
A Volume for the Reliability Shelf	Jake Ansell Mulhim Al-Doori	37

# A Volume for the Reliability Shelf - Reliability Data Analysis

*by Jake Ansell and Mulhim Al-Doori*

## Introduction

It has always been assumed that software would be added to ASL through the efforts of individuals or groups. However it would not be wise if no opportunity were given for others beyond the authors to comment at an early stage on the material and the functions developed.

In this article a brief description of the functions which are contained within the *Reliability Data Analysis* volume is given. A beta-test version of the software is to be produced in the near future. Before this though it would be helpful to have the views of others on the functions included and those they feel should be added.

The software discussed is part of a larger project which is the development of an Expert System for Reliability Data Analysis. The Expert System is being developed in APL. A subsequent article may describe the details of the Expert System.

APL is a natural language for this volume since most of the algebra is linear in form, either based on raw data or on the ordered statistics. Hence for the most part the routines follow the mathematics without the need for great sophistication. There is though the need for iterative procedures, especially Newton-Raphson-based approaches for estimation.

The data analysis volume is assumed to be part of a larger shelf where there would be routines for Reliability Analysis and Modelling. Fault tree programmes might be written based on APL routines for logic manipulations for analysing systems. Also a Stochastic Model could be added following work done at Swansea under the supervision of Alan Hawkes.

In the future it may be that the Data Analysis volume might share code with related shelves, such as a Survival Analysis Shelf or a more general Medical Statistics Shelf. This emphasises that the software described is only a first attempt, and not the definitive model for the volume. Contributions are

welcomed and as authors we would be delighted to see the shelf grow. The names used for functions might equally transform with suggestions from others.

At present we see the possibility of six sections to the volume. These are:

- Elementary Analysis
- Graphical Analysis
- Distributions
- Covariate Analysis
- Growth Models
- Dependency Analysis

## Elementary Analysis

As with the bottom shelf this is seen as a collection of routines which have been found to be useful in the analysis of reliability data, especially in the initial stages of such an analysis. It augments the Bottom Shelf. For example the coefficient of variation (*COEFVAR*) and the index of dispersion (*INDIS*) are added since they indicate whether the data is under- or over-dispersed compared to the Poisson Process. This aids in model selection. Also included is a function to calculate the cumulative mean time to failure, since again many practitioners find it a useful measure.

For chronological data Ascher has frequently suggested the need to test for trend. Hence the volume contains two of the standard trend test measures, the Laplace Trend (*TRENDLP*) and the MIL-HNBK-189 (*TRENDMIL*) test.

## Graphical Analysis

Graphical analyses are often used in Reliability. Engineers find them comprehensible and informative. They can give rise to parameter estimates, but more appropriately can indicate the validity of the model. It is assumed that the volume will provide just the values to be plotted and not the plotting routines, which should exist within ASLGRAPH. Hence the volume will produce X-Y coordinates for plotting.

The routines cover the standard plots to assess lifetime distributions such as Weibull (*WEIBPLOT*), Hazard (*HAZPLOT*) and TTT (*TTTPLOT*) plots. These plots often shed light on the appropriateness of a distribution but may also indicate possible problems with the data. A function also exists to produce a Duane Growth Plot (*DUANEPLOT*). This plot is often used to assess whether time between failures is increasing (good) or decreasing (bad).

Beyond these there will be functions to produce plots for more sophisticated analysis, such as residual plots for covariate analysis (see Covariate section) and for examining dependency such as Walls' method (*WALLSPLOT*).

## Lifetime Distributions

Obviously looking into the future there should be a separate shelf for distributions, covering the common distributions. Already, though, certain aspects of some distributions have been covered by the Bottom Shelf. In the interim it seems sensible to include some routines associated with lifetime distributions in this volume. (A lifetime distribution is generally taken to be a distribution which takes only positive values.) Hence included in this volume are routines for parameter estimation for a number of lifetime distributions.

Given the reliability context these functions allow for the possibility of censoring, currently right-hand censoring. The distributions considered are the Exponential, both one- and two-parameter forms (*EXPEST*, *EXP2EST*), the Weibull, two- and three-parameter forms (*WEIBEST*, *WEIB3EST*), the Gamma (*GAMMAEST*) and Log Gamma (*LGAMMAEST*).

Since it is also possible to obtain empirical estimates for distributions we have included Kaplan-Meier for reliability function (*KMEST*) and Nelson-Altschuler for cumulative hazard function (*NAEST*).

## Covariate Analysis

Cox's Regression Model, referred to as the Proportional Hazard Model (PHM), has led to a resurgence of interest in modelling lifetimes through the use of covariates. There is a wide range of possible models. The current volume contains functions for two models: PHM and Weibull Regression. These functions include estimation of parameters with approximate standard deviations, (*PHMBETA*, *WEIBREGBETA*), likelihood values, (*PHMLIKE*, *WEIBREGLIKE*), and residuals, (*PHMRESID[I]*, *WEIBREGRESID*). (The *[I]* indicates more than one residual available.)

Specific formulations of other models are covered by ASLREG either directly or by transformation, for example Gamma, Log Gamma and Log Normal Regression. Hence it seems inappropriate to repeat them in this shelf.

## Growth Models

It had been assumed that as technology develops, systems and components will improve especially in the context of hardware. The more pragmatic will realize this is not the case. In Reliability, however, there are a number of models for describing growth which assume positive growth for hardware: the Duane Growth Model, ARINC and others. Most of these models should contain a governmental health warning. For some of the models there is more than one estimator for the parameters of the models. For the Duane Growth model it is possible to use either graphical methods or Crow's estimate for Non-Homogeneous Poisson Process, (CROWEST).

Beyond hardware models there are a stream of software models such as Jellinski-Moranda (JMEST) and Littlewood-Verrall (LVEST). These models equally have drawbacks, but the Littlewood-Verrall model has the advantage that improvement is only expected at every failure, not required.

## Dependency Analysis

There is a wide range of possible approaches, for example Engineering yields the beta-Factor model and its extensions, whilst Statistics provides a rich medium for analysis of dependency. Currently there is not a single approach regularly used and hence at this stage we have decided not to include any specific methods except the graphical approach to analysis suggested by Walls (*WALLSPLOT*).

## Conclusions and Comments

This article has briefly covered what is contained within the volume. It is hoped to release the volume for beta-test later in the year; it will be supplied with limited documentation. If you have comments on the material so far included, or wish to suggest further functions which might be added, please write to me at the address given in the editorial on page 30.



# A Review of Glimpse

by **Mulhim Al-Doori**, *Department of Management Systems Sciences,  
University of Hull*

GLIMPSE (Generalised Linear Interactive Modelling with PROLOG and Statistical Expertise) is a knowledge-based front-end system which provides a task language as part of a high-level interface to GLIM 3.77. It arose out of an ALVEY funded project to investigate the use of logic programming techniques and tools to develop front-ends to large software packages such as GLIM. The aim of GLIMPSE is to provide assistance and optional guidance in the application of Generalized Linear Modelling. For the purposes of the review GLIMPSE was implemented on a SUN workstation but unfortunately a fast version of sigma-PROLOG was not available. This caused the sessions to be considerably lengthened at all stages.

The front-end uses the logic-based Expert System shell APES which is written in LPA Sigma-PROLOG. APES provides a declarative dialogue in which the user can supply statements which are checked against the relationship within the rule base. It yields explanation facilities following the computational steps in an analysis. There are three main parts to GLIMPSE; the TRANSLATOR, the ABSTRACT STATISTICIAN and GLIM (which is the standard Glim 3.77 program).

The TRANSLATOR translates GLIMPSE commands to GLIM commands. It provides a richer medium than standard GLIM. No longer is there a need to either proceed through tedious definition of variables or use so many MACROS which we had come to love. Hence a plot may be specified for a transformed variable without transforming the variable. Obviously this is an advance on GLIM, but one questions why it was not already incorporated in the GLIM package. There is no real need for the PROLOG front-end to deliver this facility.

Yet since it is within the present structure one pays the penalties of having to be more confined. Data entry, at least on the initial attempt, can be both tedious and hazardous. There are apparently several stages to pass through. Data Input (DI), Data Definition (DD), Data Validation (DV) etc. A number of individuals have commented that by the time you entered the data your patience has been so eroded that one abandons the session.

This leads to the problem of saving material once entered – a far from easy task. There are no clear instructions on how one opens files required or really on *how* to save material. It may be that some of these faults will be resolved by increasing familiarity with the software, but it is unfortunate that such seems difficult in a supposedly *Expert System*.

A log is maintained, both of the user's commands and of the session's output. This slowed considerably the loading of software on the implemented system, taking about 2 minutes to load at the start of every session.

The ABSTRACT STATISTICIAN is portrayed as an eye overlooking the analysis. The concept is rather puzzling, as will be discussed later. The ABSTRACT STATISTICIAN contains the distillation of the advice/expertise from some very senior statisticians who are named in the publicity material. At any stage one can ask this learned object advice on what to do next. Advice is supplied and this can then be questioned. This is the central plank of the system, since it is *the* expert part of the system.

It does not take long, however, before one starts to question the advice given. Two examples illustrate at least minor concerns about the package. The test of linearity is the fitting of a quadratic, and the method used to examine the appropriateness of including a predictor variable is to fit the variable on its own and see if significant, or add the variable having fitted all other variables. Both can be questioned and it leads to the interesting game of finding data sets which contravene the simplicity of the rules.

It is difficult, though, to imagine any set of fool-proof rules which could be constructed to deal with all eventualities in regression. The main importance of the ABSTRACT STATISTICIAN is an attempt at formulating *some* rules and establishing a sketch of the plausible strategies. Hence on this ground one sees the worth of the venture as an academic exercise.

There is no real need to discuss GLIM itself since that has been reviewed many times before. It is possible to intersperse basic Glim commands with GLIMPSE dialogue, but this is likely to lead

to problems as the ABSTRACT STATISTICIAN will not know about anything done this way. In practice, when you return to GLIMPSE the workspace is put back as it was before, so any changes made in GLIM will be lost.

On to GLIMPSE's performance. Firstly obtaining a consistent pattern of results is difficult within the range of the software choices. Having apparently followed the same steps more than once I have obtained different results even when following the texts closely. One would also suggest that it is essential to have the Reference Manual to hand throughout your initial analyses. I needed to refer to it constantly. This seems to defeat the main aim and characteristic of an Expert System, that it should be user friendly and hence aid the user through an analysis from beginning to end. The command structure and the use of the commands is not clear.

Generally one has reservations about the use of this software. Familiarity might help to resolve the difficulties but the cost to the user seems prohibitively high to gain the familiarity. The experience of other users who 'dipped into' GLIMPSE, whether or not they were already GLIM users, was that the terminology was intimidating and the

explanatory messages from the system were unintelligible.

This brings me to the last and possibly the most important question – the market for the software. GLIM was a fine idea but the software lacked user-friendliness. This prohibited the casual user from choosing GLIM, therefore GLIM never really expanded beyond the academic circle.

If GLIMPSE were more user-friendly, it might have achieved a larger market for GLIM. It still seems, though, to be user-unfriendly based on our experiences. The front-end that has been built is too cluttered. Hence the wider audience has been excluded again. But if GLIMPSE is designed for the competent statistician, why should the competent statistician need dangerously simplistic advice?

To end on a more positive note, I would regard GLIMPSE as a solid step in the direction of future statistical software. It is obviously a sound piece of academic software, and has developed some useful rules and attempted to develop strategy for analysis. Therefore it is addressing the correct questions. Currently it still has to develop to meet the requirements of the market place before it becomes a marketable piece of software.